Optimisation Models and Algorithms for Workforce Scheduling and Routing

by José Arturo Castillo Salazar, MSc

Thesis submitted to The University of Nottingham for the degree of Doctor of Philosophy.

December 2015

Abstract

This thesis investigates the problem of scheduling and routing employees that are required to perform activities at clients' locations. Clients request the activities to be performed during a time period. Employees are required to have the skills and qualifications necessary to perform their designated activities. The working time of employees must be respected. Activities could require more than one employee. Additionally, an activity might have time-dependent constraints with other activities. Time-dependent activities constraints include: synchronisation, when two activities need to start at the same time; overlap, if at any time two activities are being performed simultaneously; and with a time difference between the start of the two activities. Such time difference can be given as a minimum time difference, maximum time difference, or a combination of both (min-max). The applicability of such workforce scheduling and routing problem (WSRP) is found in many industries e.g. home health care provision, midwives visiting future mothers, technicians performing installations and repairs, state agents showing residences for sale, security guards patrolling different locations, etc. Such diversity makes the WSRP an important combinatorial optimisation problem to study. Five data sets, obtained from the literature, were normalised and used to investigate the problem. A total of 375 instances were derived from these data sets. Two mathematical models, an integer and a mixed integer, are used. The integer model does not consider the case when the number of employees is not enough to perform all activities. The mixed integer model can leave activities unassigned. A mathematical solver is used to obtain feasible solutions for the instances. The solver provides optimal solutions for small instances, but it cannot provide feasible solutions for medium and large instances. This thesis presents the gradual development of a greedy heuristic that is designed to tackle medium and large instances. Five versions of the greedy heuristic are presented, each of them obtains better results than the previous one. All versions are compared to the results obtained by the mathematical solver by using the mixed integer model. The greedy heuristic exploits domain information to speed the search and discard infeasible solutions. It uses tailored functions to deal with each of the time-dependent activity constraints. These constraints make more difficult the solution process. Further improvements are obtained by using tabu search. It provides moves based on the tailored functions of the greedy heuristic. Overall, the greedy heuristic and the tabu search, maintain feasible solutions at all times. The main contributions of this thesis are: the definition of WSRP; the introduction of 375 instances based on five data sets; the adaptation of two mathematical models; the introduction of a greedy heuristic capable of obtaining better results than the solver; and, the implementation of a tabu search to further improve the results.

Publications

1. Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2012). A survey on workforce scheduling and routing problems. In *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)*, pages 283–302, Son, Norway

2. Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2014a). Computational study for workforce scheduling and routing problems. In *ICORES 2014 - Proceedings of the 3rd International Conference on Operations Research and Enterprise Systems*, pages 434–444

3. Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2014b). Workforce scheduling and routing problems: literature survey and computational study. *Annals of Operations Research*, doi: 10.1007/s10479-014-1687-2

4. Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2015). A greedy heuristic for workforce scheduling and routing with time-dependent activities constraints. In *ICORES 2015 - Proceedings of the 4th International Conference on Operations Research and Enterprise Systems*, pages 367–375, Lisbon, Portugal. INSTICC, Scitepress

5. Laesanklang, W., Landa-Silva, D., and Castillo-Salazar, J. A. (2015). Mixed integer programming with decomposition to solve a workforce scheduling and routing problem. In *ICORES 2015 - Proceedings of the 4th International Conference on Operations Research and Enterprise Systems*, pages 283–293, Lisbon, Portugal. INSTICC, Scitepress

Acknowledgements

This research programme would not have been possible without the financial support provided by: the University of Nottingham through the International Research Excellence Scholarship awarded to overseas students; Mexico's National Council for Science and Technology (CONACYT) via the scholarship (SICOB 214017); and The United Kingdom's Engineering and Physical Sciences Research Council (EPSRC) via the LANCS initiative (EP/F033214/1).

I am deeply grateful to my main supervisor Dr. Dario Landa Silva for his patience, guidance and encouragement throughout the doctoral programme. In particular, for his help in securing all the necessary funding and for always being willing to review my work at very short notice. His support went beyond his academic duties.

I would like to thank as well my second supervisor Dr. Rong Qu for her insightful comments and suggestions which improved the quality and clarity of my work. My gratitude goes also to my viva voce examiners Dr Rhydian Lewis (external) and Dr. Ender Özcan (internal). Their invaluable comments were included in the final version of this thesis.

My gratitude goes towards the academics who shared their data, experience and insights, Prof. Greet Vanden Berghe, Prof. Volker Nissen, Prof. David Murray Ryan and Matias Sevel Rasmussen. Special thanks to Prof. Patrick De Causmaecker for receiving me at Kulak in Kortrijk, Belgium for a short research visit.

The unique environment of the ASAP research group provided me with all the support both at an academical and personal level to complete this programme. Thanks to the many fellow PhD researchers with whom I shared this time at Nottingham, you helped this adventure to be a more enjoyable one. Among them: Rodrigo, Wasakorn, Haneen, Maria, Jonatã, Özgür, Urszula, James, Anas, Ahmed, Juan Pedro, Stefan, Monica, Mina, Imo, Raras, Binhui, Osman and Peng. Equally important, I would like to acknowledge the support provided by the administrative staff at the school of computer science, in particular Deborah Pitchfork and Christine Fletcher.

I could not have finished this programme without the constant support and motivation of my mother, Ana Bertha, thanks for always being there regardless of time and distance. Thanks to my sister, Ana Laura, for keeping me updated with the events at home. And, I would particularly like to thank Karolina for being that special person in my life whom I can always rely upon.

Contents

Chapter 1

Introduction

This thesis is the product of four years of research on optimisation models and algorithms that can be applied to the scheduling and routing of employees. This first chapter introduces the background and motivation of the research programme. In addition, it provides an overview of the remaining chapters and discusses the contributions that this research makes to the field.

Workforce scheduling and routing refers to scenarios in which a group of skilled employees need to complete a series of activities. These activities are based at geographically different locations, thus requiring the employees to travel across the locations. As a result, adequate division of work between the employees is really important but often difficult to achieve by human planners. Creating a plan that assigns a subset of activities to different employees indicating the sequence of activities and starting time requires many considerations that depend on the nature of the scenario. For example, not every employee is qualified to perform every activity. The distance and time spent when travelling between locations differs depending on the means of transportation used by each employee. Employees' preferences regarding which activities to complete could be taken into account. If it is not possible for all activities to be completed due to an understaffed workforce, prioritising which activities to complete first, might be required. The previous description of the problem, although abstract, fits many real world scenarios such as: home care, home health care, field engineers, security guards patrolling, community midwives allocation, estate agents showcasing properties, and so on. The difference between traditional scheduling and routing problems in the literature is the human factor, i.e. the employees, who introduce variation in skills and preferences.

1.1 Motivation and Scope

The decision to pursue this area of research was taken after a seminar at The School of Computer Science at The University of Nottingham. The seminar was hosted by the Automated Scheduling, Optimisation and Planning (ASAP) research group. The ASAP research group carries out multi-disciplinary research into mathematical models and algorithms for a variety of real world problems (www.asap.cs.nott.ac.uk). The seminar was given by Professor Greet Vanden Berghe on the topic of hard combined combinatorial optimisation problems . The presented problem, one of several, referred to the difficulty encountered when trying to roster employees that are required to travel in order to perform certain tasks over a weekly period (Misir et al., 2011, 2015). Rostering or personeel scheduling is the process of building timetables for staff members within an organisation in order to satisfy the demand of good or services (Ernst et al., 2004b). Prof. Vanden Berghe argued that this combined problem of rostering and routing of labour was often needed by organisations and that more research should be directed to address it. The experienced on personnel rostering obtained by the author of this thesis previous the starting of his doctoral studies motivate him onto taking the challenge proposed by Prof. Vanden Berghe.

After an initial review of the literature, it was identified that before including a rostering component which included constraints that cover more than one working day e.g. maximum hours per week, it was necessary to schedule a single day of activities. This requires knowing which employees are on shift on that day and focusing on their routing while matching their skills with those required by the activities. The rostering component was meant to be included at a later stage during the research programme. Such stage did not occur and the entire research programme was spent on “daily problems that require scheduling and routing of employees”.

Once the scope of the research had been decided, the first attempt to name the problem was proposed as Flexible Mobile Workforce Scheduling and Routing (FMWSR). The name focused on two characteristics of the employees involved in this type of scenarios: flexibility, in terms of skills, preferences and working times; and mobility, travelling to perform activities at customer locations, as opposed to residing in an office throughout the day. The term was later changed to Workforce Scheduling and Routing Problem (WSRP) for two reasons: to reflect the closeness of the problem to the VRP, and because WSRP conveys the wider applicability of the problem to real world scenarios. We hope the term becomes accepted by researchers working on this research topic.

1.2 Thesis Contributions

The research presented in this thesis contributes to the understanding of the workforce scheduling and routing problem by providing:

1. Five data sets, discussed in Chapter 4, comprising of 375 WSRP instances. The sources of the data sets include similar problems that can be modelled as a WSRP. The data sets available online at: `http://www.cs.nott.ac.uk/~jac/dataset.html`

2. The adaptation of two mathematical models from the literature in order to tackle WSRP. The first model can be used when the requirement is that all activities presented in an instance need to be assigned to an employee schedule. The second model considers the fact that sometimes all activities cannot be allocated and tries to minimise the penalty incurred for the unassigned activities.

3. The modelling of teams using synchronisation constraints and virtual activities. In addition, a reduction in the number of variables in the underlying network of the MILP model is presented.

4. The design and development of a deterministic greedy heuristic for the WSRP. The heuristic provides good valid results for large instances using significantly less time than the mathematical solver. It relies on domain specific information and focuses on tackling complex activities first.

5. A tabu search implementation using OpenTS to tackle WSRP. The TS uses tailored moves to handle complex activities and adapts the tabu tenure after a number of non-improving iterations have passed. Insights into the parameter settings of the TS are provided.

6. Overall, the benchmark results obtained through the mathematical solver, the greedy heuristics, and the TS provide a starting point for comparison of new or adapted solution methods for WSRPs.

1.2.1 Publications

The following published articles have resulted from the research work presented in this thesis:

1. Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2012). A survey on workforce scheduling and routing problems. In *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)*, pages 283–302, Son, Norway

   This publication introduces the workforce scheduling and routing problem by providing a survey of recent literature. The main characteristics of every WSRP are also defined along with many others found during the survey process. Chapters 2 and 3 cover the majority of the findings.

2. Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2014a). Computational study for workforce scheduling and routing problems. In *ICORES 2014 - Proceedings of the 3rd International Conference on Operations Research and Enterprise Systems*, pages 434–444

   The publication covers the findings of Chapter 5 with regard to the integer linear programming model on a subset of the data sets.

3. Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2014b). Workforce scheduling and routing problems: literature survey and computational study. *Annals of Operations Research*, doi: 10.1007/s10479-014-1687-2

   The journal paper includes a revised version of the survey by matching each of the main characteristics of the WSRP to some industry sectors. It also presents the complete data set used in the research programme. Finally, it contains benchmark results when using a mathematical solver to tackle the MILP model. Chapters 4 and part of Chapter 5.3 are based on the work presented in this publication.

4. Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2015). A greedy heuristic for workforce scheduling and routing with time-dependent activities constraints. In *ICORES 2015 - Proceedings of the 4th International Conference on Operations Research and Enterprise Systems*, pages 367–375, Lisbon, Portugal. INSTICC, Scitepress

   This publication presents the fourth version of the greedy heuristic designed to tackle WSRP. It uses the benchmark results of publication 3 to compare and evaluate the heuristic. Part of Chapter 6 is included in this manuscript.

5. Laesanklang, W., Landa-Silva, D., and Castillo-Salazar, J. A. (2015). Mixed integer programming with decomposition to solve a workforce scheduling and routing problem. In *ICORES 2015 - Proceedings of the 4th International Conference on Operations Research and Enterprise Systems*, pages 283–293, Lisbon, Portugal. INSTICC, Scitepress

   This publication presents a decomposition technique for tackling WSRP. The technique is not part of this thesis. My contribution to this publication is in the definition of the WSRP and support on the initial modelling of the problem. The publication is included as it is a clear extension on the work I initiated during my research programme. This is another publication on which the work described in Chapter 2 has been part of the contribution.

1.2.2 Additional Planned Publications

Two additional publications are being considered based on the results presented in this thesis.

1. Heuristic methods for the WSRP.

   The planned publication will present the last version of the greedy heuristic described in Chapter 6 and another heuristic developed in collaboration with a visiting researcher, Dr Federico Alonso Pecina.

2. A Tabu Search approach for solving the WSRP.

   This paper will present the tabu search implementation for solving WSRP. It will be based on the results obtained in Chapter 7.

1.2.3 Other Contributions

The early findings of this research programme influenced the start of a Knowledge Transfer Partnership (KTP) [1] between the University of Nottingham and Webroster Ltd. The KTP started in January 2014 and is due for completion in January 2016. The project aims to improve home care workforce utilisation by developing an adaptable software optimisation engine that solves any workforce management scenario that includes both rostering and routing.

[1] see http://info.ktponline.org.uk/action/details/partnership.aspx?id=9240

The topic covered by this thesis has influenced three other research programmes at doctoral level. Currently, the first research programme is focusing on additional exact approaches via mathematical modelling with decomposition techniques for the WSRP (doctoral research programme of Wasakorn Laesanklang started in June 2012). The second research programme is investigating the multi-objective nature of WSRP (doctoral research programme of Rodrigo Pinheiro started in January 2013). Finally, the third research programme is focusing on evolutionary computation methods for the WSRP, particularly genetic algorithms (doctoral research programme of Haneen Algethami started in March 2013).

1.3 Thesis Overview

This section provides an overview of the contents of the remaining chapters of this thesis.

1.3.1 Chapter 2

This chapter is dedicated to defining the research problem. As stated in the motivation section, the WSRP has characteristics that can be applied to many real world scenarios. The WSRP's main characteristics are discussed in this chapter. In addition, other features encountered in similar problems are also reviewed. Finally, the WSRP is presented as a hard combinatorial optimisation problem through an example of the rapid growth of the search space.

1.3.2 Chapter 3

A review of the literature is performed in this chapter. The term WSRP is presented in this thesis although a great deal of similar work, often with different names, in different sectors has been discussed in the literature for some time. This chapter relates the WSRP to some of those already established problems. Furthermore, concepts required for the rest of the thesis are also discussed.

1.3.3 Chapter 4

This chapter presents five data sets gathered from different sources within Europe, including Belgium, Denmark and the UK. The source and the original format of the data sets are discussed. In addition, the changes to the five data sets in order to generate a total of 375 WSRP instances is also discussed. The generated instances are used for experiments in Chapters 5, 6, and 7.

1.3.4 Chapter 5

In this chapter two mathematical models are described. The first is an integer programming model (IP) that focuses on trying to assign all activities within a subset of the instances. The experiments only consider two data sets derived from the vehicle routing literature. A state of the art mathematical solver, Gurobi, is used to tackle the subset of instances. The solver is unable to provide integer feasible results for almost half of the instances, thus forcing the consideration of alternative models. The second is a mixed integer model (MIP) that considers the case when some activities cannot be performed due to an understaffed and unskilled workforce. The MIP model introduces a new objective function that includes a penalty for leaving activities unassigned among other considerations. The mathematical solver is used to obtain optimal, if possible, or feasible solutions that are used as a benchmark when comparing results in the remaining chapters.

1.3.5 Chapter 6

This chapter presents a deterministic greedy heuristic designed to obtain fast, good and valid solutions for larger instances. The greedy heuristic is discussed starting from its original design, inspired by a bin- packing representation. Four more improved versions of the heuristic are also discussed. The improvements of each version are: 1) broadening the search; 2) introducing specialised functions to address complex activities, i.e. activities having temporal dependencies on others; 3) search improvements through the creation of an index-type structure; and, 4) incorporating a branching-type process to evaluate multiple solutions.

1.3.6 Chapter 7

This chapter uses the solution structure and tailored functions defined in Chapter 7 to create a set of neighbourhood moves. The moves are then integrated into a metaheuristic implementation, Tabu Search (TS). The TS was implemented using OpenTS, a Java Tabu Search framework part of the COIN-OR library. The use of TS improved the quality of the results for some instances. The features of the TS implementation are discussed. In addition, the configuration of the TS parameters is addressed.

1.3.7 Chapter 8

The final chapter presents a recapitulation of the most important aspects of the research contributions. It also summarises the results obtained by the solution method presented in this thesis. Finally, this chapter proposes areas for future research in the study of the WSRP.

Chapter 2

Workforce Scheduling and Routing

2.1 Problem description

Workforce scheduling and routing problems (WSRPs) refer to the scheduling of employees to a series of geographically scattered activities. Employees have different skills, which influence the activities they can perform. Activities require particular skills in order to be completed appropriately. It is expected that activities required are matched to the skills of employees who perform them. The aim of these activities is to benefit the clients of the employee organisation. These clients typically are based in multiple locations. As a result, employees need to travel to different locations in order to perform each of the activities assigned to them. When the distance between locations is significant, it is common to restrict assignments to regions. Regions can be defined by clustering according to various criteria, e.g. geography or priority. Employees use diverse means of transportation e.g. walking, private vehicle, public transport, bicycle, etc. to move across sites. Travelling time is considered employees' working time thus any reduction in it results in employees potentially performing more activities. Therefore, the workforce (set of employees) can be seen as the provider of services for clients. Each activity is given a time window, a range of time by which it needs to start. Depending on the type of industry, time windows can be strict or flexible; this depends on what is stipulated in the contractual terms between the organisation and its clients. Activities might require more than one employee for its completion, thus leading to the arrival synchronisation of two or more employees. In some cases, the order in which activities are performed matter, such scenarios create *time-dependencies* between activities. The dependencies could mean that one activity needs to start or finish at the same time as another activity or that there should be a certain time difference between the start (or completion) of related activities. The working time of an employee is represented by two values start and end time. The working time of employees should be respected when assigning activities. Otherwise, the organisation incurs an additional expenditure i.e. extra time, which in many cases is paid at more than the normal rate. Every employee is assigned a subset of activities, each employee's schedule considers the duration of each activity and the travel time required to travel between locations. The main objective is to perform as many activities as possible with the given workforce adhering to the planning horizon.

The previous WSRP description is encountered in many industries, perhaps with a different name for employees e.g. nurses, carers, security guards, engineers, etc. and different name for the activities e.g. tasks, services, jobs, works etc. but they all refer to the same abstraction. Often, within the thesis an activity is referred as a visit. Therefore, the terms activity and visit can be considered interchangeably throughout the thesis. For example technicians visiting customers to perform installations (or repairs) of special equipment e.g. broadband installation, satellite television antenna setting. In such a scenario, a customer books a time slot depending on his availability and expects the technician to arrive within that time slot in order to complete the installation. That is the case of internet providers who send qualified engineers to install routers for their customers. Another example is carers assisting elderly citizens within a district. In this case, carers' visits last for a certain amount of time during which help with one or many tasks is provided. Tasks include help with bathing, cooking or doing groceries. If we replace the carer by a qualified nurse then the range of activities changes and the support now is not only addressed to elderly people but perhaps to anyone recovering from surgery. Other cases, involve the patrolling of building facilities by security guards. The guards arrive at a location and perform a round, a vigilant walk, in the surrounding area before moving on to another location. There are numerous other examples of employees that need to travel to different locations to perform their duties e.g. salesmen attending customer demonstrations, midwives assisting future mothers and newly born babies, estate agents travelling to showcase properties to prospective buyers, handymen performing repairs to several households in a day, etc.

There are some real world problems that have similar characteristics, but they are not considered WSRP in this thesis. One example is the case of the family of dial-a-ride problems. In dial-a-ride problems, a group of users request transportation between two locations (origin and destination) from a group of drivers (fleet of vehicles). Although the dial-a-ride problem involves routing of the vehicle fleet to several locations to pick up clients, there are conceptual differences with respect to the WSRP. The main difference is that in dial-a-ride the activity is transporting the clients from the origin to the destination. There is not significant work for the drivers at each of the locations apart from picking up and leaving the customer. Another difference, the fleet of vehicles used for transportation is most of the times considered homogeneous. There is no matching of skills between the activity and the employee (driver) performing it. A classic dial-a-ride problem is the routing of taxis to several customers' request for transportation. Another example is the parcel delivering problem in which a group of messengers pick up and/or deliver parcels to different addresses. Similar to the dial-a-ride problem there is no significant duration to the activity at the different locations and there is no requirement for skill matching. Both the dial-a-ride and parcel delivery problems are well studied in the literature covering scheduling and routing techniques (Raff, 1983; Cordeau and Laporte, 2003; Parragh et al., 2008).

As the human component, employees add greater diversity of restrictions to WSRP. Some example of restrictions found in WSRP are: heterogeneous skills among the workforce; different working times in terms of duration and shifts; employees' preferences regarding which activities to perform or with whom to work in situations requiring them to work in teams; and recipients' preferences overriding those of the employees.

2.2 Main Characteristics of WSRP

2.2.1 Time windows

A *time window* indicates the time by which the activity needs to start. Time windows are commonly given by two values: an earliest starting time α and a latest starting time β. As a result, employee(s) performing an activity with an specific time window needs to adhere to its time restrictions. Time windows can also refer to finishing time, although starting time is the most common use. The difference given by $\beta - \alpha$ determines the flexibility of the time window. In some cases there is no flexibility, i.e. $\beta - \alpha = 0$, which creates an exact starting time, also known in the literature as exact time window (Eveborn et al., 2009, pg. 27). In such cases the activity needs to start at an exact time in order to comply with the constraint. Exact time windows are reported to be limiting in the literature since they do not provide scope for variation. The opposite to exact time windows is not having time windows defined at all. In this case there is no explicit minimum starting time and maximum starting time. Nevertheless, in reality it means that α value is potentially equal to the start of the planning horizon and β to the end of it minus the duration of the activity, otherwise the completion time is after the end of the planning horizon. Activities with no given time window are not constrained by a specific starting time as long as they are started and completed during the planning horizon.

Time windows are also used to indicate an employee working time. Every employee might have a different working time depending on his contractual arrangement. It is expected that all activities he performs are within his time window. In some cases employees' time window can be overridden in order to perform additional activities outside employees' working hours. Such decision comes at an extra cost, i.e. overtime payment, and when possible organisations try to avoid it, unless there is no other choice.

2.2.2 Transportation modality

Transportation modality refers to the means of transport that employees use when travelling between locations. Different transportation modalities provide different travelling times, i.e. walking to a location is expected to take more time than driving to it. Other external factors can influence travelling time such as traffic, and road closures. In some WSRP scenarios there is a range of means of transport, e.g. in home care, carers travel with their own car, by, public transport and walking. The type of transport can change during an employee route, e.g. in home health care nurses often use their own cars to move across the city or villages, but once at the destination they could park the vehicle at a central location and walk to visit several customers within the surrounding areas. If the cost of transportation is included in the objective function then clearly the means of travel affects the scheduling of employees.

2.2.3 Start and finish locations

Employees' start and finish locations differ across WSRP scenarios. Perhaps the simplest one is where all employees start and finish at the same location. Such arrangement is similar to the vehicles in the traditional vehicle routing problem (VRP). Another alternative is for each employee to start and finish their working day at their homes. This could have some advantages, for example if employees' homes are scattered across all activities' locations then assigning employees to visit those close to their homes reduces travel time. Similarly, the last activity of the day could be the nearest to the employee's home. Finally, a combination of the same starting location but different finishing locations seem to work for some home health care scenarios. An advantage of requiring all employees to start at one location i.e. main office or headquarters, is that they can be informed of and address any last minute changes to the schedule. In some cases it is necessary to go to the main office for replenishment. After performing the last activity, employees could be allowed to go home straight away without needing to return to the main site. There are many factors that influence the start and finish locations of every member of the workforce. Some examples are: flexible arrangements; whether the travel time from home to the main office and back at the end of the day is considered payable; if employees use the company vehicles which need to be returned to the main site for maintenance or simply for insurance purposes, and so on.

2.2.4 Skills and qualifications

Skills act as filters for employees performing activities. In many scenarios employees cannot perform an activity unless they are trained to do so, therefore prohibiting their assignment is necessary. In some cases every member of the workforce can perform any activity. This is called an homogeneous skilled workforce, as opposed to an heterogeneous skilled workforce composed of employees with different skills that can only accomplish certain activities. Industries such as management consultancy and health care rely on their employees having a diverse set of unique skills to cover their clients' needs. In such industries, it is not cost effective to train all employees with the same skills as it often takes years to become a specialist in a field. Skills within the workforce can be cumulative or not. Cumulative skills allow two medium-skilled employees to perform the job of a highly-skilled employee since both employees' skills will accumulate. Other scenarios do not allow this accumulation of skills, i.e. an employee does or does not have the skills to perform an activity. The matching between employees' skills and activities can or cannot exist, i.e. treated as a boolean value. Alternatively, the matching can be given within a range, i.e. using a percentage value. Boolean matching tests whether an employee can perform an activity given his skills. Using a range provides a scale within skills for example, a senior level manager can perform the activities of a junior level manager but the opposite is not true.

2.2.5 Service time

Service time is equivalent to the duration of the activity when performed by an employee. It can vary depending on the skills and experience of the employee who perform the activity, but for the majority of WSRP scenarios it is previously estimated. There is an assumption regarding the service time/duration of activities in a WSRP. The service time of any activity should be less than the planning horizon, and when possible, employees should perform more than one activity. If the service time is long enough that it only permits employees to perform one activity in a working day then the routing component of the problem is gone and the problem becomes a task allocation. Expected duration time is often used to plan the average number of activities employees must perform during the planning horizon. The service provider company is normally paid for this time.

2.2.6 Connected activities

Connected activities refer to the time dependency that activities have with other activities, e.g. two activities might need to start at the same time, or the start of an activity may depend on the completion of another one. These type of time-based restrictions are common in vehicle routing problems. The connected activities constraints as defined by Rasmussen et al. (2012) can be of five types: synchronisation, overlap, minimum difference, maximum difference and min+max difference. Synchronisation requires two or more activities starting at the same time or perhaps finishing at the same time. Overlap means that at any time two or more activities need to be performed simultaneously. Minimum difference is a certain time that must pass from the start of an activity to the commencement of another one. Maximum time gives a deadline by which an activity must start in relation to another activity's start time. Combining minimum difference and maximum difference produces a min+max difference. A min-max difference creates an additional time window on the dependent activity which also relies on the independent activity to set a starting time. Connected activities are not present in all WSRP but when they appear they tend to make the problem harder to solve. Rasmussen et al. (2012) argue that having connected activities instead of time windows gives the search more flexibility. Nevertheless, in some cases the use of both time windows and connected activities constraints is necessary.

2.2.7 Teaming

Team formation may be necessary due to the nature of the work to be carried out (Li et al., 2005). Some activities require more than one employee when performed. If team members remain unchanged throughout the planning horizon, the team as a whole can be scheduled as a single person, since for all activities the team travels together. If the team is only formed to tackle one activity, then synchronisation of the arrival of team members to the activity's location is necessary. The second option provides more flexibility and less cost since once the team finish the activity every member can travel to a different location and continue on to other visits.

2.2.8 Clustering

Clustering refers to the grouping of a subset of the activities which have something in common. Clustering might be used for various reasons. One reason is employees' preferences regarding not travelling more than a certain number of miles away from home, in such case a cluster of possible activities is created for that employee. Another reason is when employees are designated to work only in certain regions that are close to an organisation site. Clusters can be used to reduce the number of activities requiring planning and solving, thus reducing the difficulty of the original problem.

2.3 Other Characteristics of WSRP

The eight characteristics mentioned in Section 2.2 are the principal ones of any workforce scheduling and routing problem. Perhaps with the exception of connected activities and clustering the other six are always present. Additionally, there are other features in this type of problems. The following paragraphs describe these additional features.

2.3.1 Multiple trips

This characteristic comes from the routing component of the problem. A trip is the set of visits that an employee is scheduled to perform before going back to his end destination. *Multiple trips* allow the employee to go back to the main site before starting another set of visits. This is not a very common feature but helps to model scenarios where an employee has a split shift, e.g. in the morning performing one set of visits and in the evening another set of visits, leaving the afternoon free.

2.3.2 Preparation time

In many scenarios preparation time is considered as part as the service time (duration) of the activity. However, in some industries monitoring the time after arriving to the location and before starting the activity is important in order to reduce it. Examples of activities that are performed during preparation times include unloading materials, setting up equipment, etc.

2.3.3 Driving restrictions

Driving restrictions only apply when employees use vehicles for prolonged periods of time. It is a common requirement, and in some cases based on law, that after some time of continuous driving there should be a rest period e.g. in the UK after driving for 4.5 hours a break of 45 minutes is required.

2.3.4 Preferences

Preferences should influence the assignment of employees to certain activities. Depending on the sector, preferences can favour the employees or the recipients of the activity to be performed. Employee preferences are normally introduced as a benefit, a symbol that the organisation takes employees' wishes into account in an attempt to retain them. This is particularly important in industries with high employee turnover. Employee preferences can be in reference to the location of visits, the time of visits, the type of activity, etc. Recipient preferences are considered part of the service agreement provided by the organisation, e.g. in home care an elderly woman may prefer to be assisted by a female carer when bathing. Recipient preference can be as restrictive as to indicate which employee should perform the job. Preferences are sometimes difficult to satisfy and in the majority of scenarios are modelled as soft constraints.

2.3.5 Heterogeneous Shifts

Shifts indicate the availability of employees during certain periods of time through the planning horizon. In some cases it is assumed that employees are available to work all the planning horizon. In other cases, employees can start working at different times but are expected to remain busy until the end of the shift which could or not match the end of the planning horizon. Employees with split shifts add difficulty to the scheduling process as they have two starting times and two finishing times.

2.3.6 Rostering restrictions

When the planning horizon covers more than one day, there are inevitably some periods of inactivity. Scheduling a week of activities can be tackled by solving each day as a different problem. However, this approach does not guarantee finding the optimal solution to the weekly problem. Moreover, there are additional constraints that appear in such scenarios, e.g. employees cannot work more than 40 hours a week, or after working a late shift employees cannot be assigned a morning shift the following day. Restrictions of such nature, are known as Rostering constraints and need to be considered when planning for a prolonged period of time (Ernst et al., 2004b,a).

2.3.7 Shared Transportation

Sometimes it is convenient to use one vehicle to transport multiple employees. It does not necessarily mean that employees are on the same team or that the activities they perform require more than one person. It is used as a means of reducing expenditure, i.e. when travelling costs have to be reimbursed. This requires synchronisation of employees arriving to the location of the vehicle.

2.3.8 Break scheduling

Having breaks during a prolonged working period might be a legal requirement. Breaks can be taken at the discretion of the employee after/before performing an activity and should not involve travelling time. In other words, travelling time cannot be counted as a break. In some industries breaks are scheduled as part of the plan of activities, e.g. home care. There could be flexibility regarding the way breaks are taken, e.g. employees could chose to take two 30-minute breaks during a day instead of one hour break.

2.3.9 Number of workers

In WSRPs the number of employees is limited. Nevertheless, the problem can be modelled as having an infinite number of employees. In such cases, all activities must be completed regardless of the number of employees used. The objective then is to complete the activities with the minimum number of employees. It is assumed that if additional employees are required, the planner could use casual staff. Casual employees are commonly used for periods of high demand for activities.

2.3.10 Overtime

Overtime is defined as the time employees continue working once their shift has ended or once they have surpassed the number of hours per week that they are contracted to work. Overtime provides flexibility to cover extra activities or to compensate for unplanned employee absences. Overtime incurs extra cost, e.g. could equal pay at 1.5 times or even double the normal rate.

2.3.11 Planning levels

The planning level relates to the duration of the planning horizon. Therefore, a time period is associated with each planning level. WSRPs could be defined on a monthly, weekly or daily basis. Planning for a day may affect the week, e.g. if an employee works most of his hours on Monday and Tuesday his availability might be restricted for Friday. In cases like home health care allocation of resources are made at a weekly level, but details are planned on a daily basis.

2.4 Relation to the Vehicle Routing Problem

The routing component of the WSRP considers many different variants of the classical vehicle routing problem (VRP). In VRP the objective is to minimise the distance travelled by a set of vehicles when visiting a number of customers at different locations. Each customer must be visited only once by a vehicle and all visits should be performed. All vehicles start and end at the same location (depot). In VRP as in WSRP activities are spread across multiple locations that require vehicles travelling between them. Variants of the VRP are: the addition of time windows on visits' start time (VRPTW); visits are classified as "pick-up" when goods are collected at a certain locations and "delivery" when the previously collected goods are given to the recipient (VRPPD); capacities on the vehicles carrying goods (CVRP); presence of multiple depots where vehicles can return after performing visits (VRPMD); and multiple trips when replenishment is necessary forcing vehicles to go back to the depot (VRPMT). For more VRP variants refer to Toth and Vigo (1987); Desrochers et al. (1990); Golden et al. (2008)

2.5 WSRP as a combinatorial problem

Workforce scheduling and routing is a combinatorial optimisation problem. Finding the optimal assignment of routes to employees to complete all activities satisfying all restrictions is a challenging and difficult task. A simple approach for small-size instances is via enumeration. Enumeration analyses all possible employee routes and chooses those complying with all restrictions and offering the best objective function value.

Complete enumeration in workforce scheduling and routing is not an option for medium to large size problems due to the number of possibilities available that require consideration. The size of a WSRP is determined by the number of activities and the number of available employees. A medium size WSRP in this thesis has around 50 activities and 10 employees whereas a large size has more than 100 activities and 25 employees or more. A route is a sequence of visits to locations where activities are required. The number of possible routes is only affected by the number of activities. Table 2.1 and Figure 2.1 show the number of possible routes (R_n) to consider in a WSRP as a function of n (number of visits/activities). For example if there is only one visit A to perform, then there are two routes to consider: the first one includes activity A and the second one is the empty route {A},{}. For two activities A, B there are five routes to consider {A},{B},{A,B},{B,A},{}. Notice that it is necessary to consider the order of activities. Hence {A,B} is a different route than {B,A}. Finally, the empty set needs to always be considered as an additional route as it indicates a route where no activities are assigned. The number of possible routes R_n for n activities can be obtained using Equation (2.1).

$$R_n = \sum_{k=0}^{n} (n!)/(n-k)! \tag{2.1}$$

This combinatorial explosion so far does not consider the employees. All routes may need to be evaluated for every employee to ensure the optimal solution has been found. Many routes in a given problem are invalid due to time related (time windows and time-dependencies) or skills based constraints.

The WSRP can be considered an *NP-hard problem* i.e. Non-deterministic polynomial time hard, as it is a combination of two NP-hard problems. The personnel scheduling problem (Brucker et al., 2011)and the vehicle routing problem with time windows (Lenstra and Rinnooy Kan, 1981).

# Activities n	R_n	Factorial(n)
1	2	1
2	5	2
3	16	6
4	65	24
5	326	120
6	1957	720
7	13730	5040
8	109601	40320
9	986410	362880
10	8049701	3628800

Table 2.1: Shows number of routes R_n that can be generated based on the number of activities (n) present in a WSRP. Many routes would be infeasible due to time related constraints, but they are still part of the search space of the problem. As a means of comparison the factorial function is provided to demonstrate that the search space grows at a similar rate.

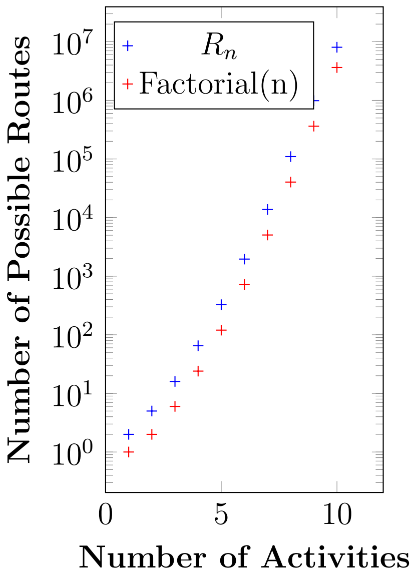

Figure 2.1: Shows the values of Table 2.1 in a graphical form. A comparison between the increase of possible routes to consider in a WSRP based on the number of activites n (+,in blue) and, the factorial function for the same n value. The rate of growth is similar.

Chapter 3

Background

The chapter provides background information on Workforce Scheduling and Routing Problems. In the first section, a review of solution methodologies is performed. The purpose of this review is to establish some common terminology which will be used throughout the thesis. The second section focuses on related work to the WSRP. A review of personnel scheduling and vehicle routing with time windows (VRPTW) as independent problems is provided. Following the description of both personnel scheduling and VRPTW a series of scenarios that present a combination of both problems are presented and discussed. Such problems are identified as potential WSRP as they present the characteristics defined in Chapter 2. In the last section, every discussed methodology is match to several surveyed papers as a way of informing the reader which approaches have been used when tackling WSRP.

3.1 Solution Methodologies

Using a similar approach to Bechtold et al. (1991) and Ernst et al. (2004a) the solution methods are grouped in three categories: exact methods including mathematical programming and constraint programming among other; and, heuristic algorithms including tailored heuristics and metaheuristics. A third category is included which covers hybrid methods as any method which uses both of the previous categories.

3.1.1 Exact Methods

Exact methods are capable of finding optimal solutions whilst guaranteeing their optimality (Burke and Kendall, 2014). Exact methods often require more computational time than heuristics, especially when large instances are being considered. Among other exact methods we can find: linear programming, constraint programming, and branching & bound methods. These approaches are sometimes referred as classical techniques (Dowsland, 2014).

3.1.1.1 Linear Programming

A liner programming problem is an optimisation problem that can be modelled entirely by the use of linear expressions. There are two parts in an optimisation problem an objective function and a set of constraints/restrictions on the solution to the problem. In other words, both the objective function and constraints need to be linear expressions based on a set of decision variables. A feasible solution is one that satisfy the constraints defined in the problem. Due to their linear representation these problems can be solved by well known methods such as the Simplex type methods. The version of a linear program where the all variables are restricted to integers values is called integer linear program (IP). Moreover if in an IP the only values allowed for the variables are the yes or no type then it is known as a binary linear programming. If there is the case where some variables are continuous and other integers the problem becomes a Mixed integer linear program (MIP). (Dowsland, 2014; Brucker and Knust, 2006)

3.1.1.2 Constraint Programming

Constraint programming (CP) is an exact approach based on logic implications. When used to tackle optimisation problems, the set of variables must be linked by a set of constraints. The variables can only take their values from a finite set of integers. The constraints could be represented mathematically or through symbolic operators. Solving a CP requires interleaving two process, a propagation and a search. The previous with the aim of finding valid solution for the problem. The propagation stage consist of reducing the variables that wont lead to feasible solution. The search stage is triggered after the propagation one. The objective of the search is to fix inconsistent values in the variables. The search uses a tree based procedure that reduces the problem into subproblems (Talbi, 2009).

3.1.1.3 Branch & Bound, Branch & Price and Branch & Cut

A way of guaranteeing that an optimal solution has been found is to analyse all possible solutions. Such process is called enumeration. For small problems enumeration is a feasible approach. Enumerating the solutions often helps to understand the structure of the problem. When enumerating it is common to use a tree based structure that represents all possible solutions. The tree could potentially hold all different possibilities of variable configurations. The root node of the tree holds branches for every variable depending on their finite number of values. The leaf nodes, i.e. those without children, in the tree represent final values that cannot be branched further. The strategy to search the tree could yield different results. The strategy needs to be defined at the beginning of the optimisation process. Common strategies include a depth-first approach which explores an specific area until a terminal node, i.e. leaf, is found and subsequently it backtracks to the nearest junction. Another strategy, known as breadth-first, explores the same level of the tree and whilst doing so, it is able to prune sections of the tree which given their configuration could not lead into feasible solutions. A depth-first strategy tends to find feasible solutions quickly but it neglects regions of the tree which might have better ones. A breadth-first strategy consumes a lot of memory resources but can compare across the tree and facilitates the removal of dominated subsolutions. (Lawler and Wood, 1966; Mitten, 1970; Hillier and Lieberman, 2010)

As the size of the problem increases, the size of the tree that contains all possible solution grows explosively. Branch and Bound aim to reduce the number of nodes to be analysed in the tree whilst still maintaining optimality. In the case of large problems, the algorithm is better if performing branching only in selected regions of the tree. The regions that are bounded, hence the name, for two values: an upper bound and an estimated lower bound. The branch and bound helps to prove that some partial solutions represented in the tree structure will not lead to optimal solutions hence discarding them from the search this process is called pruning.

Branch and Price refers to the combination of branch and bound and column generation methods. It consists on decomposing the original combinatorial optimisation problem into two types of sub-problems. A master problem and a pricing problem. It is a method commonly used to solve large inter programming models and mixed integer ones. (Feillet, 2010; Danna and Le Pape, 2005)

Branch and Cut uses branch and bound in combination with cutting planes techniques to gradually reduce the search space of the problem. Cutting planes iteratively refine a feasible set by adding linear constraints that satisfy all feasible integer points but violate the current fractional value within the tree structure. (Mitchell, 2002; Martin, 2001)

A methodology that has been very useful to tackle WSRP is branch and price. Branch and price refers to using a branch and bound approach with column generation (Barnhart et al., 1998; Feillet, 2010). The advantage of using column generation is that the problem can be relaxed and solved with a reduced set of columns, which might not be an exhaustive enumeration of all possible routes for every employee, but at any time provides a solution if it exits. In the literature, the personnel scheduling constraints side of the problem is commonly solved by heuristics to generate columns. On the other hand, the routing component can be tackled via branching. Kallehauge et al. (2005) showed that the problem formulation can be decomposed into a master problem and a pricing problem. The master problem is a set partitioning problem and the subproblem a series of shortest path problems with resources constraints (Irnich and Desaulniers, 2005; Feillet et al., 2004).

Models applied to VRPTW have also been aplied to WSRP, in particular multi-commodity network flow models with time windows and capacity constraints. When using branch and price, many authors have modelled the master problem as either a set partitioning problem or as a set covering problem. There is not much difference between these two. In the first one, each customer is in one route only, whereas in the second one, more than one route could visit the same customer location.

3.1.2 Heuristics Algorithms

In this section a description of metaheuristics methods used in the tackling of workforce scheduling and routing is presented. For each metaheuristic a brief overview is performed and then reference to relevant work in the literature is provided.

Metaheuristics are high-level search methods which guide and influence other heuristics to increase their chances of finding good valid solutions in the search space. They offer a framework structure that is applicable to any domain which makes them non-problem specific. Metaheuristics use domain specific knowledge in their implementation(Osman and Laporte, 1996; Glover and Laguna, 1999; Voßet al., 1999).

Metaheuristics can be classified according to more than one criteria. Among the most common ones are the following: origin or inspiration of the algorithm therefore there are nature-inspired and non-nature inspired metaheuristics. Number of solutions simultaneously, single point search or population based. Single point search act over only one solution trying to improve it with every iteration. Population based have many solutions which evolve by combining characteristics of the solutions to pass them to the next generation. Classification based on the objective function nature which could be static or dynamic. Number of neighbourhoods, most metaheuristics use one neighbourhood but the possibility of using more than one in order to change the topology of the space search is a way of differentiating metaheuristics. The final classification is whether memory structures are used or not. Memory-less metaheuristics perform iterations based only on their current state without remembering good solutions or regions with potential to explore (Blum and Roli, 2003)

3.1.2.1 Trajectory Methods

Trajectory methods refer to metaheuristics that focus on a direction of movement within the search space. Such a search process is seen as changes of stages in a discrete time. It all starts in an initial state and traverses the search space using a strategy until termination criteria have been achieved. The dynamic nature of the trajectory (path) depends on the algorithm, the problem representation and the problem instance. Trajectory methods are single point search metaheuristics.

Two of the most successful trajectory methods that have been applied in a wide range of application domains include simulated annealing and tabu search.

3.1.2.2 Simulated Annealing

Simulated annealing has its origins in the work by Kirkpatrick (1984) inspired by the annealing process of solids, i.e. the evolution of a solid in a heat bath in order to achieve thermal equilibrium. Given a current state of a solid with Energy E_1 subsequent states can be generated by applying perturbations. If the energy difference of the next state is less or equal to 0 then the state is accepted, if the energy difference is greater than 0, the state is accepted with certain probability ρ. Following that analogy, simulated annealing accepts deteriorations in cost with different values. The beginning of the search process with bigger values and as the search progresses only smaller deteriorations are accepted. Similar behaviour can be achieved by the use of a probability distribution which assigns low probability to large increments and high probability to small increments. The acceptance of worsening solutions is meant to escape local optima. The acceptance level is controlled by a temperature parameter T which is decreased as the search continues.

3.1.2.3 Tabu Search

Tabu Search (TS) is a popular metaheuristic when tackling combinatorial optimisation problems. TS acts on a single solution and tries to improve it by using memory structures. Memory allows the algorithm to keep track of explored regions in the search space and avoid cycles (going back to recently explored solutions). Memory also permits escaping local optima and exploring other regions in the search space. Memory can be of two forms: short memory and long memory.

Short memory is implemented via a tabu list of forbidden solutions. It is often costly to store entire solutions in the tabu list, thus it is more common to store attributes of such solutions or the moves used to generate the solutions. At every iteration the best neighbourhood solution is kept and its attributes/moves are added to the tabu list. In subsequent iterations such solutions (attributes/moves) are restricted. The length of the tabu list is called tabu tenure, and it decides how many iterations attributes/moves are forbidden. Eventually, an attribute/move stops being tabu and is taken into consideration to generate new solutions. The tabu tenure dictates whether the algorithm explores a region (small tenure) or it moves to other regions in the search space (big tenure) allowing diversification. The tabu tenure can be altered during the search process. By doing so, the TS algorithm could be more robust as it can control intensification stages with diversification ones. An intensification stage focuses on a single region in the search space as it might be promising for obtaining better results. Diversification is required when local optimum has been achieved and the algorithm requires to escape the current region and continue exploring the search space.

Long memory refers to the information collected throughout the search process and not only while some attributes/moves are restricted in the tabu list. Information regarding the number of attributes/moves that have been applied (frequency); the attributes/moves used in the last K iterations (recency); how good/bad solutions have been in regions of the search space (quality); and the influence of a certain decision during the search, e.g. changing the tabu tenure. The four principles of long term memory described earlier (frequency, recency, quality and influence) allow the algorithm trajectory to be strategically guided.

One additional concept in TS is the notion of an aspiration criteria. This refers to accepting a solution even when such solution's attributes/moves are marked as tabu. The most common aspiration criteria is when the solution obtained is better than the current best. The stop criterion can be either a number of iterations or computation time. If all improving moves are marked as tabu, i.e. no more moves are allowed, this can also be used as a termination criterion.

3.1.2.4 Other Trajectories methods

There are other trajectory-based methods such as variable neighbourhood search (VNS) which introduces the concept of more than one neighbourhood structure. Changing the neighbourhood structure allows the metaheuristic to escape local optima. Guided local search alters the objective function during search thus changing the landscape of the search space. By doing so the algorithm escapes local optima and it is able to continue exploring the search space. Changing the objective function is achieved by the introduction of penalty and regularisation parameters. Iterated local search applies local search to an initial solution until a local optimum is found at that point the solution is perturbed (changed) and another stage of local search is applied. Perturbing the local optima helps to escape from it.

3.1.2.5 Population-based methods

Population-based metaheuristics act upon multiple solutions at a time, which allows the exploration of different regions of the search space simultaneously. The population is manipulated as time passes to focus on parts of the search space. The manipulation mechanism depends on the nature of the algorithm, e.g. in ant colony optimisation it is the pheromone track that ants leave as they explore promising regions. Many analogies with natural phenomena have inspired population-based method.

3.1.2.6 Ant Colony Optimisation

Ant Colony Optimisation (ACO) is a metaheuristic inspired in the behaviour of ants. Ants are able to find the shortest path between food sources and the colony. At early stages scout ants explore the surrounding in almost a randomised way. As soon as the scouts find a food source they secrete a pheromone that other ants can track. Other ants will then join in consuming the recently found food source. The more ants join a path the stronger the pheromone track becomes as it is reinforced by the ants. Then as new sources of food are found and the current source is consumed the pheromone track diminishes in intensity. The track of pheromone is modelled with a parametrised probabilistic model which allows the ant to decide which track to pursue. Ants represent a single solution that is being constructed and the components of good valid solutions can be seen as the food which ants try to incorporate into their solution. Components in combinatorial optimisation problems such as the WSRP can be assignments or constraints. It depends on how the ant analogy is exploited. Ant Colony Optimisation includes Ant-Systems (Dorigo et al., 1996), Ant Colony System (Dorigo and Gambardella, 1997), and Max-Min Ant Systems (Stützle and Hoos, 2000). Although the general analogy is the same, the update of the pheromone trails differs in each of them.

3.1.2.7 Particle Swarm Optimisation

Particle Swarm optimisation (PSO) originated as a simulation on a simplified social system. Its creators wanted to graphically describe how a flock of birds moves (Eberhart and Kennedy, 1995). In PSO a solution is represented by a particle which moves towards the position of better solutions. Each particle is assigned a velocity parameter determining how fast it can move. Whilst moving to this better position other positions along the way could also be explored. Each particle keeps track of its current position and the position of its neighbours (other particles that are somehow adjacent to it topologically). The algorithm keeps track also of the best solution achieved, which changes as particles explore new regions of the search space. The tension between moving towards the local best (neighbourhood only) and the global best whilst varying the velocity at each iteration allows the algorithm to explore and find different solutions. PSO requires tuning on the following parameters: the number of particles, initial velocity and the change in velocity.

3.1.2.8 Other populations-based methods

Other population-based methods include Evolutionary Computation based Algorithms (Genetic Algorithms, Evolutionary Programming and Evolutionary Strategies) inspired by evolutionary theory (Talbi, 2009, chap. 3). These methods allow individuals to combine and mutate so as to form a new generation of better adapted individuals. The algorithms create a set of solutions which are then evolved using two process: mutation and recombination. Mutation perturbs a single solutions. Recombination takes some attributes of two or more solutions and creates a new one. The optimisation process is permitted by the introduction of a selection process which measures the fitness of a solution (individual) to pass to the next generation or to recombine with others to generate offspring. The principle follows the survival of the fittest in which only attributes that form parts of good solutions are passed from generation to generation. Mutation can be allowed and it is controlled by a mutation rate parameter which in many cases is responsible for the diversification across the search space.

When employing heuristics including metaheuristics and hyper-heuristics to solve the WSRP, there seems to be a tendency in the literature to use approaches based on swap (exchanges) and insertion operators. Depending on the method employ either memory is used to keep the best solutions so far or to remember which low-level heuristics are best applied in the stages of the search. Many solutions employ a constructive heuristic to generate a fast initial solution. There seems to be no solution method applied to different WSRP scenarios so far. Nevertheless, the operators used to generate neighbour solutions appear to be very similar in the different approaches.

3.1.3 Hybrid Methods

For the purpose of this Thesis a hybrid method is one that uses both exact methods and heuristics methods in its implementation. Hybrid methods are used when there are clear stages in the solution process and it can be identified that for example for one stage mathematical programming can be used and in a final stage a heuristic to try to improve upon the quality of the solution. The combination of each exact method and heuristic could lead to a plethora of approaches. Among the most common hybrid approaches in the literature are combining mathematical programming with any heuristic know as Matheuristics and combining constraint programming and heuristics.

3.1.3.1 Matheuristics

Most hybrid approaches try to combine the most appropriate algorithms depending on which part of the WSRP is being tackled (clustering, routing, matching skills, etc). For the routing part, it seems that the most used approaches are mathematical programming and constraint programming. This might be due to the significant advances in optimisation methods achieved recently for vehicle routing problems. Nevertheless, good heuristics methods, particularly those which provide fast initial solutions have also been employed. When matching employees to activities, the use of heuristics approaches appears to dominate.

3.2 Related Work

3.2.1 Vehicle routing problem with time windows

Given the similarities between workforce scheduling and routing problem (WSRP) and vehicle routing problems with time windows (VRPTW), researchers have successfully utilised VRPTW models and solution techniques to tackle WSRP-like scenarios. For example, home health care (Cheng and Rich, 1998; An et al., 2012; Nickel et al., 2012; Akjiratikarl et al., 2006, 2007; Allaoua et al., 2013), patrolling of security officers (Misir et al., 2011; Chuin Lau and Gunawan, 2012), engineers/technicians on field (Günther and Nissen, 2012). These previous works cover: time windows, start/end location, skills, service time and transportation mode. Other characteristics such as connected activities, teaming and clustering have been researched to a lesser extent in the WSRP literature. There are some exceptions, for example, connected activities have been considered by Rasmussen et al. (2012), while teaming has been considered in Li et al. (2005) and Dohn et al. (2009).

The routing part in many problems considered here as examples of WSRP is based on the vehicle routing problem with time windows (VRPTW). In this problem the main objective is to minimise the total distance travelled by a set of vehicles serving customers spread across different locations. Every customer must be visited once by one vehicle. Each customer specifies a time window when the visit should happen. The delivery vehicle must arrive at the location within that specified time window. If the vehicle arrives before the earliest start time specified in the time window, it must wait until the time window opens to perform the delivery (Desrochers et al., 1992; Kallehauge et al., 2005). Extensions of the VRPTW include other features such as multiple depots, multiple trips and synchronisation of vehicles.

In VRPTW with multiple depots (MDVRPTW), the fleet of vehicles is distributed across several depots and each vehicle needs to return to the same depot from which it started once its deliveries have been completed. The formulation of this VRPTW variant (Desaulniers et al., 1998; Polacek et al., 2004) is applicable to workforce scheduling and routing as it permits associating each employee starting and ending location (home) to a different depot. It is also possible for every employee to start at the same location (main-depot) but to end their working day at a different location (home), although this scenario is not covered by the original MDVRPTW.

Another extension of the VRP allows multiple trips (Brandão and Mercer, 1998), also called, VRPTW with multiple use of vehicles (Azi et al., 2010) when using time windows. In this scenario, vehicles are allowed to go back to the depot more than once during the planning horizon. It is often associated with perishable products in order to restock. In WSRP this applies to employees performing more than one trip on a day. A trip in this context involves a series of activities before going back to the main site. Sometimes employees might need to go back to the main site to replenish resources or to swap means of transportation as some vehicles might be restricted when accessing customers' locations (Brandão and Mercer, 1997).

Finally, another extension of VRPTW which is relevant to WSRP is the synchronisation of vehicles. Teaming can be modelled in the same way as two or more vehicles arriving simultaneously at a customer location (Bredström and Rönnqvist, 2007, 2008). Synchronisation is just a type of temporal precedence constraint in VRP (Dohn et al., 2011). In WSRP if a client/recipient/patient should be visited more than once per day, the order of visits might matter, e.g. the installation and calibration of an antenna dish needs completion before technicians can install a satellite TV modulator. These activities could be performed by different technicians at different times but the order must be respected.

There are many solution methods proposed to tackle the VRPTW. When using exact approaches, researchers tend to model the problem as multi-commodity network flow problems (Desaulniers et al., 1998; Salani and Vaca, 2011) or following a set partitioning/covering formulation (Bredström and Rönnqvist, 2007). Such models have been tackled using constraint programming, branch and bound, and branch and price (column generation) (Barnhart et al., 1998; Desrosiers and Lübbecke, 2005). Other researchers use hybrid methods that employ heuristics for the generation of columns within a column generation setting (Bredström and Rönnqvist, 2008) or use heuristics to improve an initial solution found with mathematical programming (Fischetti et al., 2004). Alternative approaches include dividing the problem into smaller subproblems and then attempting to obtain a global solution using the results of each subproblem. This approach does not guarantee finding the overall optimal global solution but it is sufficient if the objective is to find valid solutions quickly (Desaulniers et al., 1998; Halvorsen-Weare and Fagerholt, 2013; Landa-Silva et al., 2011; Laesanklang et al., 2015).

3.2.2 Personnel Scheduling

Personnel scheduling refers to the allocation of employees into shifts in order to satisfy the demand of work which varies over time. Personnel scheduling problems are very important for the service industries e.g. call centres, hospital wards, policemen, transportation personnel, etc (Pinedo, 2009). Baker (1976) classification of personnel scheduling problems includes: shift scheduling, days of scheduling and tour scheduling. Another classification by Bechtold et al. (1991) focuses on the solving methodology either: linear programming and heuristic based. Ernst et al. (2004b) expands further the classification and considers five stages. Every stage is independent and optional. The stages are: demand modelling which establishes the work to be performed; days off scheduling decides if an employee is present or not in any given day during the planning horizon; shift scheduling assigns employees which are available to a define working time; line of work constructions deals with constraints arising when building the shift pattern for any given period such as weekly or monthly; finally, task and staff assignments deal with the activities performed whilst employees are on shift. Ernst et al. (2004a) also provide a review of methods and techniques used in personnel scheduling. The methods are grouped into five categories: demand modelling, artificial intelligence methods (fuzzy set theory and expert systems), constraint programming, metaheuristics, and mathematical programming.

A more recent classification on personnel scheduling by Van den Bergh et al. (2013) includes related problems with regards to the setting or technical features of the problem. They surveyed a range of different personnel scheduling problems in the scientific literature and provided the characteristics of instances from the data sets used in those studies. For example for personnel characteristics, they differentiate types of availability i.e. full-time, part-time, casual. Another characteristic for the type of decisions for tasks, sequence, groups, time and other. Another characteristic includes solution techniques. They identified that some personnel scheduling problems present a combination of features with the vehicle routing problem, but such problems were not included in their classification as they recognised that it was a different research field. It is precisely these combined problems that are the focus of this thesis.

3.2.3 Workforce Scheduling and Routing

In this section some of the problems tackled in the literature that can be considered as a type of workforce scheduling and routing problem (WSRP) are reviewed. The intention is to illustrate the variety and importance of WSRP scenarios in the real-world. Each subsection focuses on a problem domain and the solution methods that have been used in the literature to tackle it. Distinction is made between *exact*, *heuristics* and *hybrid methods*.

3.2.3.1 Home Health Care

Bertels and Fahle (2006) describe home health care (HHC) as visiting and nursing patients at their home. Patients' preferences regarding the time of visit are respected as much as possible, as they should not be kept waiting. Additionally, nurses have time limitations on the number of working hours per day or the starting and ending time. In HHC, transportation modality is present when nurses travel (car, public transport or walking) to visit more than one patient. The start and end location can vary. Nurses can depart from their homes or from a central health care office, and end their day once they return home or in some cases at the last visited patient's location. A diverse set of skills and qualifications usually exists among nurses. Health care organisations often cannot afford to have nurses trained in all procedures. Therefore the use of highly qualified nurses should be restricted to tasks that demand those skills. Nursing tasks can vary in duration (service time), e.g. from a 10-minute injection to a 45 minutes for physical therapy. Time-dependent nursing activities can arise when administering medication, e.g. the first dose is applied in the morning followed by another dose three hours later. Some activities require more than one nurse at the same time, e.g. handling a person with epilepsy. In such cases, nurses can be synchronised to arrive at the location at the same time. Clusterisation is used by the organisation providing health care to avoid nurses having to travel overly long distances.

Other characteristics of HHC include nurses' preferences and shift types. Also, it is desirable to avoid changing which nurses visit particular patients because patients and nurses develop a bond that is usually good to maintain. Cheng and Rich (1998) explore the use of casual nurses. Their work does not consider different skills and qualifications but instead, they propose a matching method in which a pairing, patient-nurse, is either feasible or not. The objective in their work is to reduce the amount of overtime and part-time work employed.

HHC has been tackled mainly with hybrid approaches. For example combining mixed integer programming with heuristics for either the routing or the scheduling component (Begur et al., 1997). Another example of combining two approaches is when using constraint programming to obtain a good feasible solution and in a second stage applying a series of metaheuristics including simulated annealing and tabu search (among others) to improve the quality of the solution (Bertels and Fahle, 2006).

Among the pure heuristics methods is the application of variable neighbourhood search by Trautsamwieser and Hirsch (2011). Exact methods have also been used, particularly branch and price, using a set partitioning formulation for the master problem. The model includes real variables for the scheduling of the activities, and binary variables for deciding whether an activity is performed by a specific employee or not. The pricing problem is an elementary shortest path (Barnhart et al., 1998; Bredström and Rönnqvist, 2007). An extension of such models includes the addition of side constraints in the master problem (Dohn et al., 2008). Not all models for the set partitioning part have both real and binary variables, for example pure integer models are also used by Kergosien et al. (2009). The addition of cuts on the time windows improves the branch and price approach and turns it into a branch cut and price which has lead to good results in VRP (Fukasawa et al., 2006) and hence applied later to HHC as a result (Trautsamwieser and Hirsch, 2014)

3.2.3.2 Home Care

The home care problem, also called domiciliary care, refers to the provision of community care service by local authorities to their constituents (Blais et al., 2003; Akjiratikarl et al., 2006; Borsani et al., 2006; Thomsen, 2006; Akjiratikarl et al., 2007; Justesen and Rasmussen, 2008; An et al., 2012). The aim is to schedule care workers across a region in order to provide care tasks within a time window while reducing travel time. This problem is related to the HHC problem described earlier (Bertels and Fahle, 2006; Cheng and Rich, 1998). The difference is that HHC involves helping people for a relatively short period of time to recover after hospitalisation. Home care however usually refers to helping elderly and/or disabled people to perform their daily activities such as shopping, bathing, cleaning, and cooking, etc. (Eveborn et al., 2009). Once a person starts receiving home care support it is likely that he remains receiving such care for a long time.

Home carers usually start travelling from home to deliver support at their predefined destinations using their own transport arrangements (mixed transportation modality) and return home at the end of the day. In some cases reported in the literature, care workers do not start from their home but from a home care office as last minute changes to their schedules are possible and need to be agreed before starting the working day (Eveborn et al., 2009). In some cases, travel time is considered as work hours and hence the objective is to reduce the time used not providing care. In other cases, like the work by Dohn et al. (2008), the objective is to maximise the quality level of care service provided. Reducing cost, although important, is not usually the main objective. Dohn et al. (2008) study the problem as a variant of the VRP with time windows. Although not as much as in HHC, there are some skills and qualifications required in home care when caring for others, e.g. health and safety, handling people with dyslexia, etc. Service time is standardised and it only varies due to the experience of the carer or difficulties with the person receiving care. Time-dependent activities also exist in home care, e.g. taking a shower before grocery shopping. Teaming is usually not present as carers tend to be synchronised to perform difficult tasks, e.g. assisting a heavy person. Clustering is based on municipal borders to clearly define which authority (e.g. council, district, etc.) is responsible for each area.

Additional features of home care include prioritising visits. Usually there is not enough personnel to perform all visits in a single day. Therefore, visits are rescheduled or even cancelled in the worst case. Deciding which visit is not carried out is part of the problem. For example, it is more important to assist someone with his diabetes medication than to help another person in grocery shopping. The shift patterns are either given by contracts or expressed as preferences by carers. Many organisations emphasise respecting carers' preferences to increase staff retention. Also, tolerance on time windows to perform care tasks can vary widely, e.g. 5 minutes tolerance for critical medical activities, 15 minutes to 2 hours tolerance for support activities, etc.

Home care problems have been solved using all three exact, heuristics and hybrid approaches. Among the exact methods we find linear programming (De Angelis, 1998). Mixed integer linear programming is also used on assignment and scheduling models of home care problems. The assignment model is used when new visits are introduced and the scheduling model is used to generate weekly visits (Borsani et al., 2006). Heuristic methods include local search based on simple heuristics, metaheuristics like tabu search (Blais et al., 2003), evolutionary approaches such as particle swarm optimisation (Akjiratikarl et al., 2006, 2007) and agent-based modelling (Itabashi et al., 2006). Other methods include hyper-heuristics (Misir et al., 2010). Among all heuristic methods the solving strategy seems to be similar to generate a good initial solution followed by local improvement procedures. Common neighbourhood moves include insertion, removal and swaps to interchange both activities among workers and activities in an employee's route. The combination of a set partitioning model and a repeated matching algorithm, to find suitable pairs of employees and routes in a hybrid approach has also been used to tackle home care (Eveborn et al., 2006, 2009). Matheuristics, combine mathematical programming with metaheuristics have also been used to tackle home care (Allaoua et al., 2013).

3.2.3.3 Scheduling Technicians

Some telecommunication companies require scheduling employees to perform a series of installation and maintenance jobs. In the literature, this problem is referred to as technician and task scheduling problem (Cordeau et al., 2010), field workforce scheduling (Lesaint et al., 2003), field technician scheduling problem (Xu and Chiu, 2001), technicians routing and scheduling problem (TRSP) (Pillac et al., 2011; Kovacs et al., 2012; Pillac et al., 2013) and technician-dispatching problem (Weigel and Cao, 1999). In this sector, commitments on time to perform the jobs are enforced, resulting in strict time windows. Technicians need to carry equipment so it is common to use company vehicles to travel from one customer location to the next one. Technicians start and end at the company premises, although in some cases they are allowed to take home the company car depending on the location of the first job the following day. Technicians are often highly skilled and this can be related to their experience and training. As a result companies define levels of seniority (e.g. junior technician, supervisor, etc.) among their workforce. Those seniority levels to some extent help to estimate the service time required to complete the job. Activities tend to be independent from each other within the same day, but in a wider time frame there are some connections between them. In this scenario, teams are often formed with the aim of having a balanced set of personnel with as many skills as possible. Teaming also helps technicians to learn from each other, hence improving their performance. Companies with many branches across different regions use clustering to assign jobs to each branch when the scheduling is done centrally for all branches.

The scheduling of technicians has been solved using heuristics approaches, particularly a fast constructive heuristic to reach valid solutions (Xu and Chiu, 2001). Then, local heuristics based on destroy and repair moves are used to improve the solutions (Ropke and Pisinger, 2006). Different heuristics are used depending on the stage of the problem that is is being tackled: activities allocation to employees, skill matching, and routing (Cordeau et al., 2010). Greedy randomise adaptive search procedure (GRASP) has been successfully applied in this domain (Hashimoto et al., 2011). Moreover, evolutionary approaches like particle swarm optimisation have been reported to find good enough solutions for instances of 300 employees (Günther and Nissen, 2012).

Among exact methods mathematical programming models focusing on the nature of a diverse set of skills among the workforce is reported by Firat and Hurkens (2012). Additionally, combining constraint programming with branch-and-price is reported to obtain promising results (Cortés et al., 2014) when including information regarding the maximum number of repairs (services) per day for technicians.

Finally, a parallel Matheuristic used by Pillac et al. (2013) hybridises constructive heuristics, parallel adaptive large neighbourhood search (ALNS) with mathematical programming to tackle fictitious extended instances based on the Solomon benchmarks.

3.2.3.4 Security Personnel Routing and Rostering

In this problem, a round of visits are performed by security personnel to several customer premises in different locations over a 24-hour period (Misir et al., 2011). Many organisations outsource security guard duties only when premises are closed, while in other cases security is outsourced at all times. Round visits must be performed at the contracted time, often given as a time window. Security personnel often use vehicles to go from one location to the next one and then walk once they get to the facility but are required to check several buildings. Security guards often have their own home as the start and end of the working shift. In Misir et al. (2011) the authors mention 16 types of skills that the company records for its workforce and some visits require enforcing those skills. The duration (service time) of each visit can vary but it should be within a time window in which the visit must finish. Visits are independent from each other. Customers are divided into regions (clustering), so that security guards living nearby are assigned to each region to reduce travelling time. In this industry, contract terms vary considerably leading to many different additional constraints. Although not mentioned in the scenario, it is not unreasonable to assume that teams of two or more guards are used.

A mathematical programming approach was used by Chuin Lau and Gunawan (2012) when solving a similar problem that involved security teams to patrol different underground stations within the network. Hyper-heuristics is another method that has been applied to this problem by using two different heuristics selection methods, simple random and adaptive dynamic, followed by an improvement heuristic (Misir et al., 2011).

3.2.3.5 Manpower Allocation

The manpower allocation problem (Lim et al., 2004) refers to assigning servicemen to a set of customer locations to perform predefined activities e.g. repairs, inspections, sales, promotions, etc. The objectives are to minimise the number of servicemen used, minimise the total travel distance, minimise the waiting time at service points, and maximise the number of activities assigned. The manpower allocation problem therefore can be seen as another example of WSRP. Manpower allocation with time windows is particularly relevant since customers explicitly define when the workforce is required. There is no mention of transportation modality so it is assumed all servicemen use the same type of transport. Every serviceman starts and finishes his working day at the control centre. Skills among the workforce are assumed to be the same, making no difference on who performs the service.

There are restrictions on the number of hours each employee can work. Waiting time, the time that servicemen have to wait at a customer location before the start of the time window, is included within the service time making it vary accordingly. Li et al. (2005) add job teaming constraints, where a team is assembled at every location and work cannot start unless all members of the team have arrived. More recently, a variation of the manpower allocation problem was tackled in the context of scheduling teams to do ground handling tasks in major airports (Dohn et al., 2009). In the work by Li et al. (2005) teams are set at the beginning and do not change over the working day. Additional characteristics include teams having mandatory breaks within certain time windows, hence breaks are treated as just another visit. Three types of solution methods have been identified in the manpower allocation literature. An exact method uses integer programming, based on a set covering formulation which is solved with branch and price (Dohn et al., 2009). Metaheuristics including tabu search, simulated annealing and squeaky wheel optimisation have also been applied (Lim et al., 2004; Cai et al., 2013). Finally, Li et al. (2005) relaxed an integer programming formulation of a network flow model to obtain lower bounds. The upper bounds were obtained using constructive heuristics and simulated annealing was the main component of their solution framework.

3.3 Summary

In Table 3.3, row 1 associates each surveyed source with a domain problem mentioned in Section 3.2 while row 2 indicates the main technique used for its solution.

Table 3.1: Characteristics overview in WSRP

#	Characteristics	Begur et al. (1997)	Brandão and Mercer (1997)	Brandão and Mercer (1998)	Cheng and Rich (1998)	De Angelis (1998)	Desaulniers et al. (1998)	Blais et al. (2003)	Lim et al. (2004)	Li et al. (2005)	Akjiratikarl et al. (2006)	Bertels and Fahle (2006)	Borsani et al. (2006)	Eveborn et al. (2006)	Itabashi et al. (2006)	Akjiratikarl et al. (2007)	Bredström and Rönnqvist (2007)	Dohn et al. (2009)	Bredström and Rönnqvist (2008)	Dohn et al. (2008)	Eveborn et al. (2009)	Kergosien et al. (2009)	Cordeau et al. (2010)	Misir et al. (2010)	Rasmussen et al. (2012)	Misir et al. (2011)	Landa-Silva et al. (2011)	Salani and Vaca (2011)
1	Domains:																											
	HC Home care																											
	HHC Home health care	HHC	VRP	VRP	HHC	HC	VRP	HC	MA	MA	HC	HHC	HC	HC	HC	HC	VRP	MA	VRP	HHC	HC	HHC	ST	HC	HHC	SP	VRP	VRP
	ST Scheduling technicians																											
	SP Security personnel																											
	MA Manpower allocation																											
	VRP Vehicle routing																											

Table 3.1 – continued from previous page

#	Characteristics	Begur et al. (1997)	Brandão and Mercer (1997)	Brandão and Mercer (1998)	Cheng and Rich (1998)	De Angelis (1998)	Desaulniers et al. (1998)	Blais et al. (2003)	Lim et al. (2004)	Li et al. (2005)	Akjiratikarl et al. (2006)	Bertels and Fahle (2006)	Borsani et al. (2006)	Eveborn et al. (2006)	Itabashi et al. (2006)	Akjiratikarl et al. (2007)	Bredström and Rönnqvist (2007)	Dohn et al. (2009)	Bredström and Rönnqvist (2008)	Dohn et al. (2008)	Eveborn et al. (2009)	Kergosien et al. (2009)	Cordeau et al. (2010)	Misir et al. (2010)	Rasmussen et al. (2012)	Misir et al. (2011)	Landa-Silva et al. (2011)	Salani and Vaca (2011)
2	Solution method employed: O. Optimisation 1. MILP 2. ILP 3. INLP 4. SPP 5. MCNFP H. Heuristics 6. TS 7. PSO 8. SA 9. SWO 10. Agents based 11. LNS, VLNS, ALNS 12. Hyper-heuristics C. Hybrid	O. MILP	H. TS	H. TS	C. MILP, Constructive	O. MILP	O. INLP	H. TS	H. TS, SA, SWO	C. ILP, Constructive, SA	H. PSO	C. MILP, SA, TS	O. MILP	C. ILP, SPP, Matching	H. Agents	H. PSO	O. ILP, SPP	O. ILP, SPP	C. MILP, heuristics	O. ILP, SPP	C. SPP, Repeat matching	O. ILP	H. LNS, VLNS, ALNS	H. Hyper-heuristics	O. ILP, SPP	H. Hyper-heuristics	H. LNS, Clustering	O. MCNFP

MILP Mixed integer linear programming, ILP Integer linear programming, INLP Integer non-linear programming, SPP Set partitioning problem, MCNFP Multi-commodity network flow problem, TS Tabu search, PSO Particle swarm optimisation, SA Simulated annealing, SWO Squeaky wheel optimisation, (V/A)LNS (Very/Adaptive) Large neighbourhood search

The scheduling of employees with ‘flexible’ arrangements and ‘mobility’ is of great importance in many scenarios. Many types of personnel scheduling problems have been tackled in the literature (Baker, 1976; Miller, 1976; Golembiewski and Proehl Jr, 1978; Cheang et al., 2003; Ernst et al., 2004b; Alfares, 2004). This thesis is focused on workforce scheduling problems in which personnel is considered *flexible* (in terms of tasks and working times) and *mobile* (travelling is required in order to do the job). By *mobility* we refer specifically to those cases in which moving from one location to another takes significant time and therefore reducing the travel time could potentially increase productivity.

Chapter 4

WSRP Benchmark

4.1 Introduction

In this chapter the data set used for all experiments and computational studies is presented. The data set is the result of gathering published data from related problems that have some of the characteristics discussed in chapter 2. Once the data set was obtained, some changes had to be performed, e.g. adding constraints or information that was not included in the original source.

During the review of the literature on WSRP related topics, it was found that the majority of research in this field had been performed using generated data sets. Therefore, in order to reuse the work of previous researchers, some generated data sets needed to be considered. In addition, it was also important to obtain additional data sets based on real world scenarios in order to relating to the applicability of the WSRP in industry. In total, five data sets were obtained, two generated data sets from the VRPTW literature and three real world data sets obtained through contacting the authors of related publications.

The data sets required some adaptations because in their original form they were not compatible with each other, e.g. they had different units for distance and time. Some of the main features listed on chapter 2 were not present in all data sets, e.g. no teaming required or time-dependent activities constraints. The aim of the adaptations was to generate uniform instances that could be used when performing experiments.

4.2 Description of original data sets

The aim of this section is to provide an overview of the original data set obtained from WSRP-like problems. This section describes them as they were originally published.

4.2.1 VRPTW data set

Given the close relationship between the vehicle routing problem with time windows (VRPTW) and workforce scheduling and routing (WSRP) two data sets from the literature of VRPTW are included. The first data set is the one by Solomon which has been widely studied. The second data set is from a study of the multi-objective aspect of VRPTW (Castro-Gutierrez et al., 2011).

4.2.1.1 Solomon's data set

Solomon's data set consists of 56 instances. Each instance contains 100 visits. The instances are classified according to the duration of the planning horizon and the location of the visits. In total there are six groups of instances R100, R200, C100, C200, RC100 and RC200. The initial letter in the name of the groups refers to the type of distribution of visits used within the groups' instances. Groups R100 and R200 have a random visits distribution within the given area. Groups C100 and C200 present identifiable clusters of activities within the instances. Groups RC100 and RC200 combine random visit distribution with the presence of some clusters of visits. Groups R100 and RC100 have a short planning horizon between 230 and 240 minutes. In contrast, groups RC200, R200, C100 and C200 present a planning horizon of more than 900 minutes. In every instance there are different configurations of time windows for visits, some visits have an exact time window, others a flexible one and in some cases the time window is the same size as the planning horizon, i.e. not explicitly indicated. Solomon included instances with short service time (10 minutes) in groups R100, R200, RC100 and RC200 and long service time (90 minutes) in groups C100 and C200. There is not a defined set of vehicles per instance, because part of the objective of the VRPTW is to minimise the number of vehicles used to cover all 100 visits. Distances and travelling times are the same in absolute value. The matrix defining such values is symmetrical, i.e. the distance from location A to B is the same as from B to A. Distances are also Euclidean, i.e. the length of the line which connect to points.

4.2.1.2 Multi-objective VRPTW data set

The data set originally comes from a distribution company based in Tenerife, Spain (Castro-Gutierrez et al., 2011). It is structured in a similar way as the Solomon data set. The key differences are that distances and times values are based on information obtained via Google maps in contrast to simply euclidean values. As a result, in this data set distances and times values are different and non-symmetric. The distribution company has five types of customers, each type has its own time window profile for its required visits. The five types are: 1) Customers that are available through all the planning horizon (0-480 min); 2) Customers who prefer morning arrivals (0-160 min), afternoon deliveries (160-320 min) and late times (320-480 min); 3) Customers with similar distribution morning, afternoon and late but with a shortened time windows (130 minutes) respectively for morning (0-130), afternoon (175-305) and late (350-480); 4) Customers with even more restricted time window arrangement (100 minutes) for morning (0,100), afternoon (190,290) and late (380,480); and 5) The final group of customers consists of random selection among the previously defined time windows. Instances are grouped depending on the number of customers they contain, either 50, 150 and up to 250. In total combining three different sizes (number of activities) times five different time window profiles (1, 2, 3, 4 and 5) giving 15 instances in total within the data set.

4.2.2 Home health care data set

The origin of these instances relates to a couple of home health care real scenarios based on two Danish municipalities (Rasmussen et al., 2012). This is perhaps the most complete of the data sets in terms of WSRP's characteristics being provided. It includes skills for employees. There are four different main skills that are distributed among the carers. In addition, real average times in seconds and distances in meters are given. This is the only data set that contains preferences of both employees and recipients. Moreover, activities have an associated priority level. The priority is used because it is recognised in the industry that not all the activities can be performed in a day. Priority level might be increased as days pass without performing the corresponding activity. Finally, some instances contain time-dependent activities constraints. In total there are 11 instances in this data set.

4.2.3 Security guards patrolling data set

The data set describes the work of a set of security guards performing patrolling rounds in several locations. It has activities through out a month. The information originally comes from a Belgian company (Misir et al., 2011). The data set is divided into six districts. Each district with a range of security guards and different number of visits (patrolling round locations). It records up to 16 different skills for the security guards which provide a good range of skill matching against activities. Security guards are available 24 hours and they must start and end their work at home.

4.2.4 Technicians scheduling instance

Originally from a British Telecom Laboratories problem the only instance in this category described the assignment of 118 technicians to perform 250 dispersedly located jobs. The instance is used in the work of Günther and Nissen (2012). The distance and time matrices can be obtained following a simple formula. The duration of the jobs varies from 10 up to 513 minutes. It is the only dataset that provides average activities' duration that vary depending on the technicians' expertise. Time windows are only of three types: morning, afternoon and no preference which cover a mix of the previous two. Technicians are contracted for eight hours with different starting and finishing working times. There are 11 servicing centres and each of the technicians must start and end their working day at the designated one. Qualifications are present in this instance. Some activities can only be performed by a single employee, other simpler activities can be carried out by up to 107, thus giving a good distribution of activity-employee matchings.

4.3 Modifications to data sets

In the previous section 4.2 the original source and a brief description of every data set was provided. Although all referred problems in the data sets have WSRP features, they all differ. For example only the home health care (HHC) data set includes time-dependent activities requirements. There are no preferences being given for employees (vehicles) on both of the VRPTW based data sets. In this section the changes and additions to all data sets are described. When possible their original features are kept in order to preserve their domain characteristics. The changes are required since all instances will be used for the mathematical models and algorithms for workforce scheduling and routing problems presented in this thesis.

4.3.1 Adaptations to VRPTW data sets

4.3.1.1 Solomon's data set

Solomon's data set does not include a defined set of employees (vehicles). Therefore, a given set of vehicles is created per instance. For each of the 56 instances with 100 activities, 20 employees are assigned, i.e a fifth of the number of activities. The proportion value was decided following conversations with a service organisation within the UK home care sector, and it also matches the assumption by Bredström and Rönnqvist (2008). The organisation average visit duration is 50 minutes. Employees' shift duration is eight hours (480 minutes) with a break one break (60 minutes), or two breaks (30 minutes). Taking this into account, the mean number of visits per employee per day x is obtained by solving the equation $50x + 30(x + 1) + 60 = 480$, where $x + 1$ is the number of trips in a route including the last trip to the employee's final destination. And, 30 minutes is the average time between visits, which include travelling and idle time. The result is $x = 4.875$ rounded to 5. Additionally, two versions of each of the 56 instances were created. A version with only 25 activities and a version with 50 activities. Following the same proportion (1/5) of defined employees, instances with 25 activities are given five employees and ten employees for those with 50 activities.

The original data set did not provide any skill requirement between employees and visits. Therefore, the inclusion of a single skill against activities is introduced. The single skill is assigned to every employee with certain level of expertise. All activities required having this skill to some degree level. As a result, only some employees are able to perform all activities.

The working time for all employees is set equal to the time horizon.

Some activities were changed to require two workers instead of one. A probability was set to 0.1 for two employees and 0.9 for one.

Time-dependent activities constraints were included in the data set with the following procedure. Each activity has a 0.25 probability to be related with a time-dependent constraint with the subsequent activity. The order of activities is maintained according to the original data set. Different probabilities were assigned to each type of time-dependent constraint among the five defined in section 2.2.6. Probabilities are:

synchronisation (0.35), overlap (0.35), minimum difference (0.1), maximum difference (0.1) and min-max difference (0.1). The values for synchronisation and overlap were deliberately larger than the other three as these two are more common time-dependent constraints. As it will be discussed later in Chapter 5 scenarios where teams are required are modelled via synchronisation constraints. Validation was required when choosing among the five types in order to avoid creating time-dependent constraints that otherwise would be impossible to adhere too. For example creating a synchronisation constraint between two activities that that have non-overlapping time windows.

Employee's preferences are added in the form of a matrix defining the preference of each employee towards performing each activity. Four preference levels were created (0.2) (0.5) (0.8) (1.0) the bigger the value the stronger the preference.

4.3.1.2 Multi-objective VRPTW data set

Given the similarities with Solomon's data set, similar adaptations were performed, e.g. the introduction of a unique skill in order to match activities with employees. The same probability (0.1) was used to test if an activity should require more than one employee, i.e. a team. The addition of time-dependent activity constraints as described earlier with a 0.25 probability for an activity to be included in one of such constraints. The same probabilities were used for each type of constraint should the activity have one: synchronisation (0.35), overlap (0.35), minimum difference (0.1), maximum difference (0.1) and min-max difference (0.1). Finally, employee preferences using the same four preference levels (0.2) (0.5) (0.8) (1.0) were assigned in a random manner.

4.3.2 Adaptations to home health care data set

This data set contains most of the main characteristics. As a result, no major additions were made. Nevertheless, some minor changes were necessary. The first one included changing the time matrix, which was provided in seconds, to minutes so that it matched the rest of the data sets. The nine skills included in the data set were kept but their level was normalised to a value between 0.0 and 1.0 for both employees and activities. Finally, the priority levels, indicated by numbers in the original data set, were mapped to descriptive words with different penalty factors, low (1.0), medium (2.0), high (10.0) and urgent(20.0). The rationale for assigning such values is to give priority to urgent activities. In this sector, failing to deliver activities with high or urgent priority can have serious repercussions to the health or the recipient, hence the chosen values.

4.3.3 Adaptations to security guards patrolling data set

The original data set provides information for a month of activities and some rostering constraints. These constraints were removed. From the six districts with monthly activities, 180 instances were generated. The instances reflect each day within a month (30 days) for each of the districts, i.e. 30 days $\times$ 6 districts = 180 instances. The activities included in a day were those which required to be in that specific date or which overlap the time window of the original visit. For example, in the original data set there are requirements such as "a visit to location X must be carried out between the Wednesday 5th and Friday 7th of November". Then, only in those three day instances corresponding to the 5th, 6th and 7th the activity is included. Similarly, with employees' availability. For every day in a district, all employees available during that day are part of the instance. Such a procedure resulted in 30 independent instances for every district with some difference regarding the number of employees and activities.

Similarly to previous data sets, some activities were changed to require two employees, but in this data set the probability increased to 0.2 in comparison to 0.1 used in the VRPTW based ones. Finally, the same mechanism to include time-dependent activities constraints as described earlier is used in this data set. Nevertheless, in this case the same probability is used (0.20) for all five types of constraints.

4.3.4 Adaptation to technicians scheduling instance

Some activities were changed to require two employees instead of only one. Inclusion of time-dependent activities constraints, giving all types the same probability of being chosen (0.20).

The average time duration of every activity is used regardless of the employee that performs it. Contrary to the original data set in which an evaluation function determines given the employee skills the duration of the activities. No preferences were added to this instance.

4.4 Analyses of data sets

This section provides an analysis of all data sets. The overall analysis focuses on the following features: number of activities, number of employees, skills distribution, time windows present, duration of activities, planning horizon, activities requiring teams and distribution of time-dependent constraints. Such information helps to understand the structure of the instances after their adaptation.

In future sections and chapters each data set is referred to by the following acronyms: Solomon's data set (Sol), multi-objective VRPTW's data set as (Mov), home health care data set as (HHC), security guards patrolling data set as (Sec) and technicians scheduling (Tech). In addition some figures and tables present the symbol (#) which indicates *number of*.

4.4.1 Number of activities

The number of activities is one of many factors that provides an indication of how hard it is to solve an instance. In the VRP literature, the number of visits is constantly increased as better techniques to solve harder combinatorial optimisation problems are tested. Even, in travelling salesman (TSP), results are reported on the number of locations the salesman has to visit. Given the similarities with VRPTW, the number of activities in a WSRP relates directly with the size of the search space. As discussed in section 2.5, as the number of activities increases, the number of routes to consider grows at a rate similar to a factorial function.

Table 4.1 shows the minimum, mean, maximum, and standard deviation of number of activities in all data sets. The minimum number of activities in all instances is 25. The largest instance is 10 times bigger with 250 activities. Figure 4.1 shows the distribution of instances that have the same number of activities across all four data sets. For example, it can be noticed that the procedure used to divide the monthly district instances, in the security guards patrolling data set, produced a varied range of daily instances with 26 activities up to 210. In the figure, the distribution of the three sizes of instances regarding number of activities is shown for the Sol's data set. Three bar charts corresponding to 25, 50 and 100 activities are presented. Similarly with Mov's data set a similar pattern of three different types but this time of 50, 150 and 250 activities.

# Instances	Data Set	Min(x)	Mean(x)	Max(x)	Std. Deviation
180	Sec	26	108.04	210	53.39
168	Sol	25	58.33	100	31.27
15	Mov	50	150.00	250	84.51
11	HHC	60	101.90	153	25.01
1	Tech	250	250.00	250	-

Table 4.1: Summary of number of activities (x) in each data set. Note, Tech's data set consists of one instance therefore no standard deviation is provided

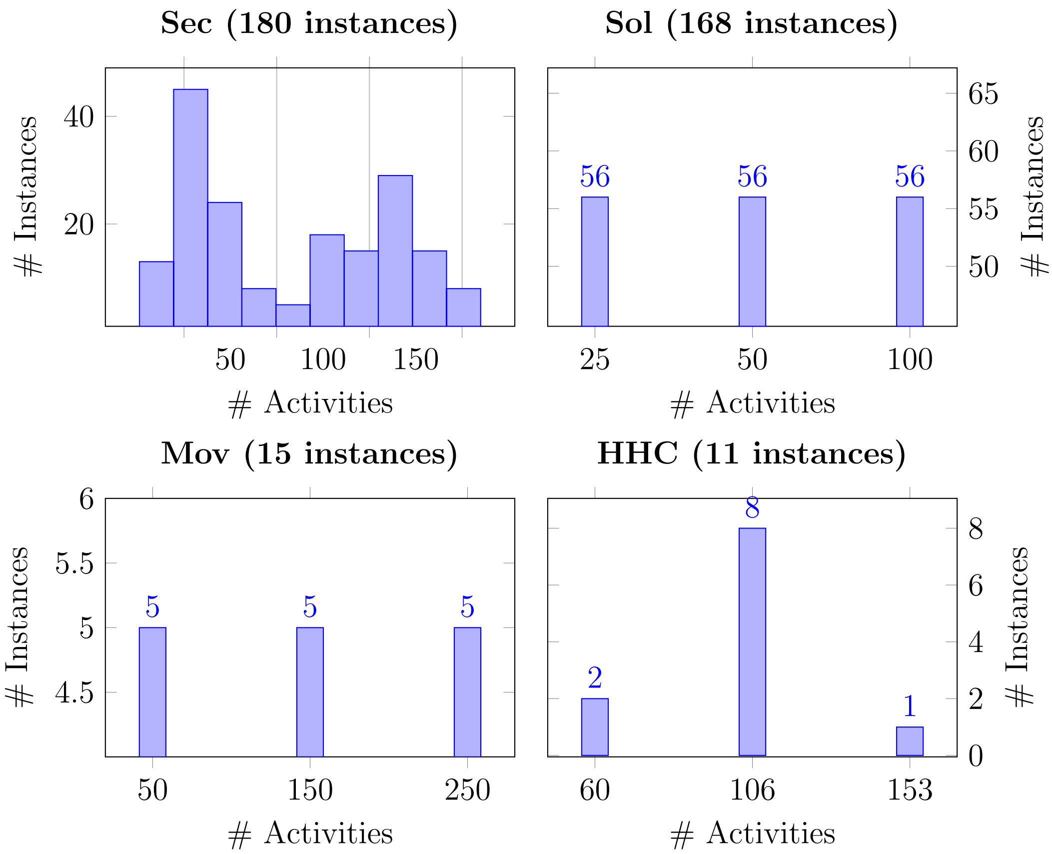

Figure 4.1: Shows the distribution of instances that have the same number of activities per each data set (Sec, Sol, Mov, HHC)

4.4.2 Number of employees

The second characteristic that directly relates to the size of the search space is the number of employees available in the workforce. Every possible route has to be tested against each employee in order to guarantee the optimum solution as a result increasing the number of employees results in more comparisons. In other related problems, such as the VRPTW, the number of vehicles (employees) is not as important as the number of visits, the main reason is that if all vehicles (employees) are seen as homogeneous then there is no need to test the all possible routes to all of the employees. Testing one is enough, since the rest are equivalent in the model. A diversified workforce in which employees cannot be easily classified in profiles, i.e. one profile applying to more than one employee, represents more difficulties for a human planner. The more information that is stored and used about employees when planning, the lower the possibilities of having "model" employees. Opposite from vehicles, ships, containers, etc. when people are involved there are always attributes of uniqueness.

Table 4.2 presents a summary of the number of employees in each data set. The smallest number of employees in an instance is five, and the largest one has 171.

The data sets assume that if an employee is present then he is available to be scheduled. In other workforce related problems such as Staff Scheduling and Rostering (Ernst et al., 2004b) one stage of the search involves deciding whether the employee could work on a particular day depending on his availability. Therefore, the number of employees available is often greater than the number reported in Table 4.2. In such cases it usually refers to the whole workforce of an organisation and not only those who are available during the planning horizon. Even though employees' availability is assumed in the data set, employees can still remain unassigned due to other factors such as skills. Employees' skills could be insufficient to perform any of the activities, making them unavailable for assignment purposes. More about employees' skills is discussed in section 4.4.3.

Figure 4.2 shows the distribution of instances that have the same number of employees. It is noticeable that the distribution is similar to the one observed in Figure 4.1. Sec data set has a more diverse range of instances with a different number of employees. In contrast both VRPTW-based data sets are just a proportion (0.20) of the number of activities.

# Instances	Data Set	Min(y)	Mean(y)	Max(y)	Std. Deviation(y)
180	Sec	6	26.63	52	13.32
168	Sol	5	11.66	20	6.25
15	Mov	38	104.00	171	56.20
11	HHC	7	9.18	15	2.08
1	Tech	118	118.00	118	-

Table 4.2: Summary of number of employees (y) in each data set. Note, Tech data set only has one instance, therefore no standard deviation is provided

4.4.2.1 Relationship between Visits and Employees

The ratio between the number of activities per employee could be used as a fairness measure when assigning activities to employees, i.e. even distribution of activities

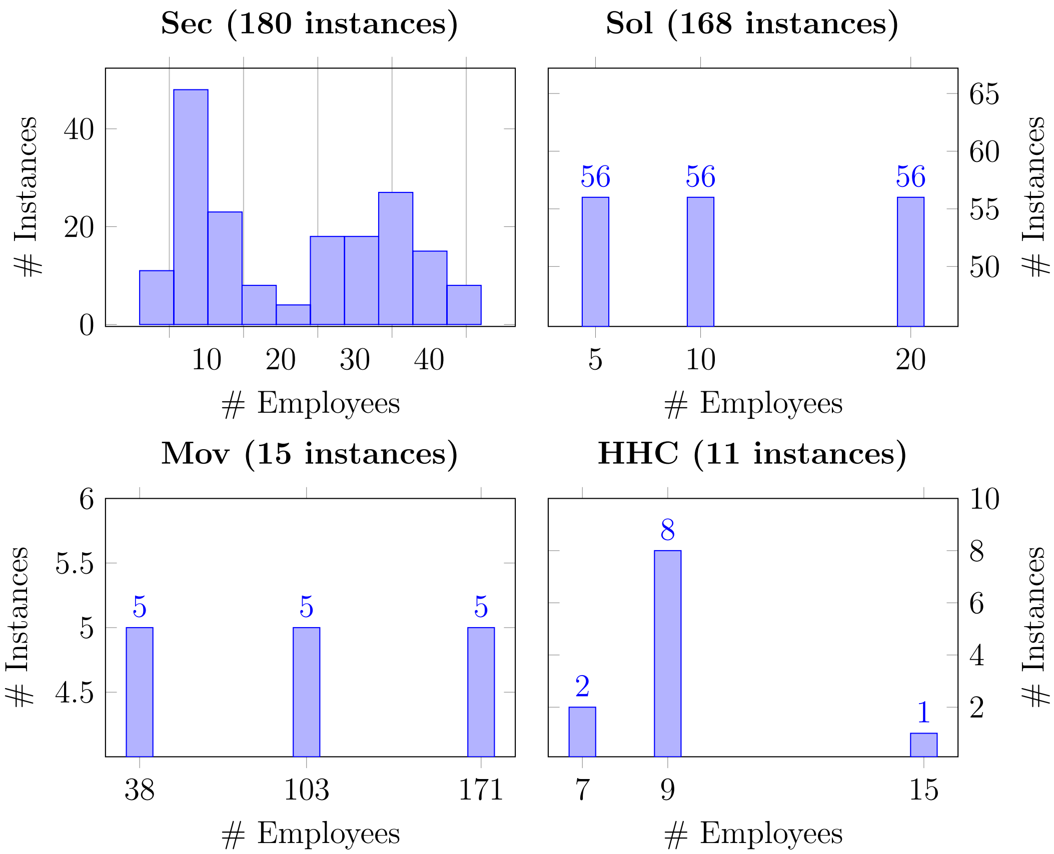

Figure 4.2: Shows the distribution of instances that have the same number of employees in each data set (Sec, Sol, Mov, HHC)

among employees. For example, if the ratio is 5:1, then the scheduling procedure could limit the number of activities assigned for every employee to 5 +/- some deviation, e.g. 1, in such case then all routes assigned to employees could have a minimum of four and a maximum of six. Table 4.3 provides the minimum, mean, maximum and standard deviation of the ratio in the instances of every data set. The ratio varies depending on the data set. HHC and Sec ratios are four and five activities per employee. On the contrary, Mov and Tech ratios, indicate less than two activities per employee. Particularly Mov, it appears to have almost one employee per activity, if that was the case, then employee-routes will only consider one location apart from the start and end destination. In such case the problem becomes a task-allocation with no routing component as every employee is required to travel to one location. Ratios close to 1.0, e.g. (1.46) for Mov combined with a small average duration of activities might indicate that some of the instances are over staffed. The original source of the Mov data set (Castro-Gutierrez et al.) confirmed that their instances have more vehicles than required. The Mov data set was not changed as having over staffed instances could be a valid scenario in the real world. HHC presents the highest ratio 11.77 which might indicate that the duration of the activities in HHC are shorter, the shift times longer or a combination of both. Later it was found that in real world home health care scenarios it is common not to complete all activities in a single day due to the reduced size of the workforce.

#Instances	Data set	Min(x/y)	Max(x/y)	Mean(x/y)	Std. Deviation(x/y)
168	Sol	5.00	5.00	5.00	0.00
15	Mov	1.31	1.46	1.41	0.06
11	HHC	8.57	11.77	11.05	1.31
180	Sec	4.00	4.42	4.07	0.00
1	Tech	1.73	1.73	1.73	-[a]

Table 4.3: Minimum, mean, maximum and standard deviation of the ratio (number of activities per employee) (x/y) for each data set is shown. [a] Only one instance so no standard deviation is provided.

4.4.3 Employees' skills

Skills restrict which activities employees can perform. A complement to the previous ratio (activities per employees) could be obtained if we consider the percentage of activities that employees can perform given their skills rather than just dividing activities/employees. Table 4.4 shows the percentage of activities that an employee can perform. Employees in HHC can perform more than 95% of the activities. In contrast, the most qualified employee in Tech can only perform a third of the activities. There are some instances in Sol and Mov which contain employees that due to their skills are unable to perform any activity (0.0%). Employees unable to perform any activity should be removed during the pre-processing stage of any solution method used. The only exception, is in the case of apprentices who by themselves have insufficient skills to do a task on their own and could follow a master to learn his trade. Figure 4.3 shows the distribution of instances that have similar percentage of activities that can be performed for the average employee when skills are considered. The percentage is shown per data set (Sec, Sol, Mov, HHC).

#Instances	Data Set	Min	Mean	Max
168	Sol	0.00%	71.66%	100.00%
15	Mov	0.00%	72.41%	100.00%
11	HHC	95.81%	98.60%	100.00%
180	Sec	33.48%	87.73%	98.59%
1	Tech	22.44%	24.51%	34.63%

Table 4.4: Percentage of the activities employees cover when taking skills into consideration. The percentage is shown for every data set.

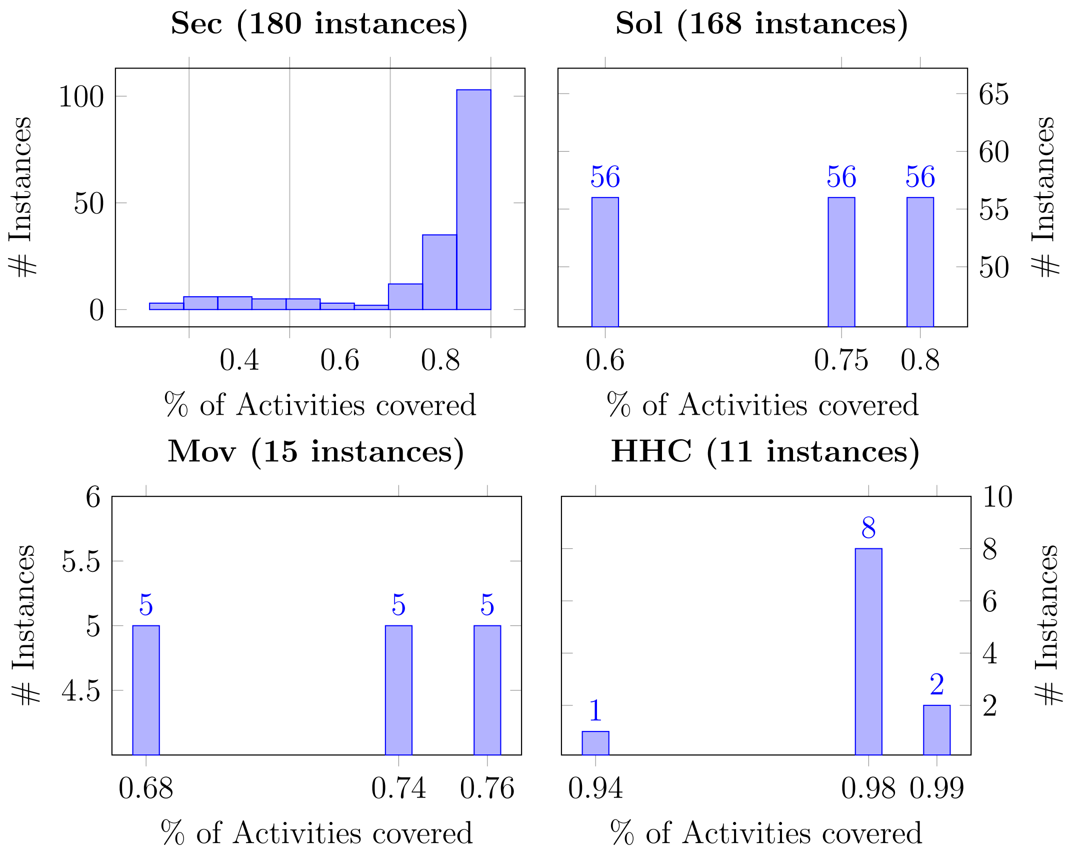

Figure 4.3: Shows the distribution of instances that have similar percentage of activities that can be performed for the average employee when skills are considered. The percentage is shown per data set (Sec, Sol, Mov, HHC)

4.4.4 Time windows

The size of the time window determines the degree of flexibility that each activity has for its start time. An exact time window could be seen as too restrictive. On the contrary, if there is no time window then the possibilities for assignment could be many. Table 4.5 shows average time window size in minutes for all data sets. Sec and Sol present the most diverse time window sizes. Sec for example has an average of minimum time window size of 117 minutes and an average maximum size of 613 minutes. Similarly, for Sec seven minutes as average of minimum time window size and 1214 minutes as maximum size. In contrast, HHC present activities which require an exact time (0 time window duration).

Figure 4.4 shows the distribution of the average time window sizes in every instance for each of the data sets. Sec presents the most diverse range. The majority of Sol instances the time window is exact, i.e. 0. There is a clear division of two average time window sizes for Mov and HHC, i.e. a small size and a big size as there is nothing in between the two extreme bar charts. For Mov the small size is 100 minutes and the big one around 400 minutes. Small size in HHC refers to below 70 minutes and big more than 100 minutes.

#Instances	Data Set	Min	Mean	Max
168	Sol	116.96	358.13	613.35
15	Mov	192.00	204.34	270.00
11	HHC	0.00	68.03	263.63
180	Sec	7.16	460.94	1214.41
1	Tech	720.00	867.50	1440.00

Table 4.5: Mean Time window size in each data set. Minimum and Maximum values are calculated per instances and then the mean of all the minimum/maximum values per data set is shown.

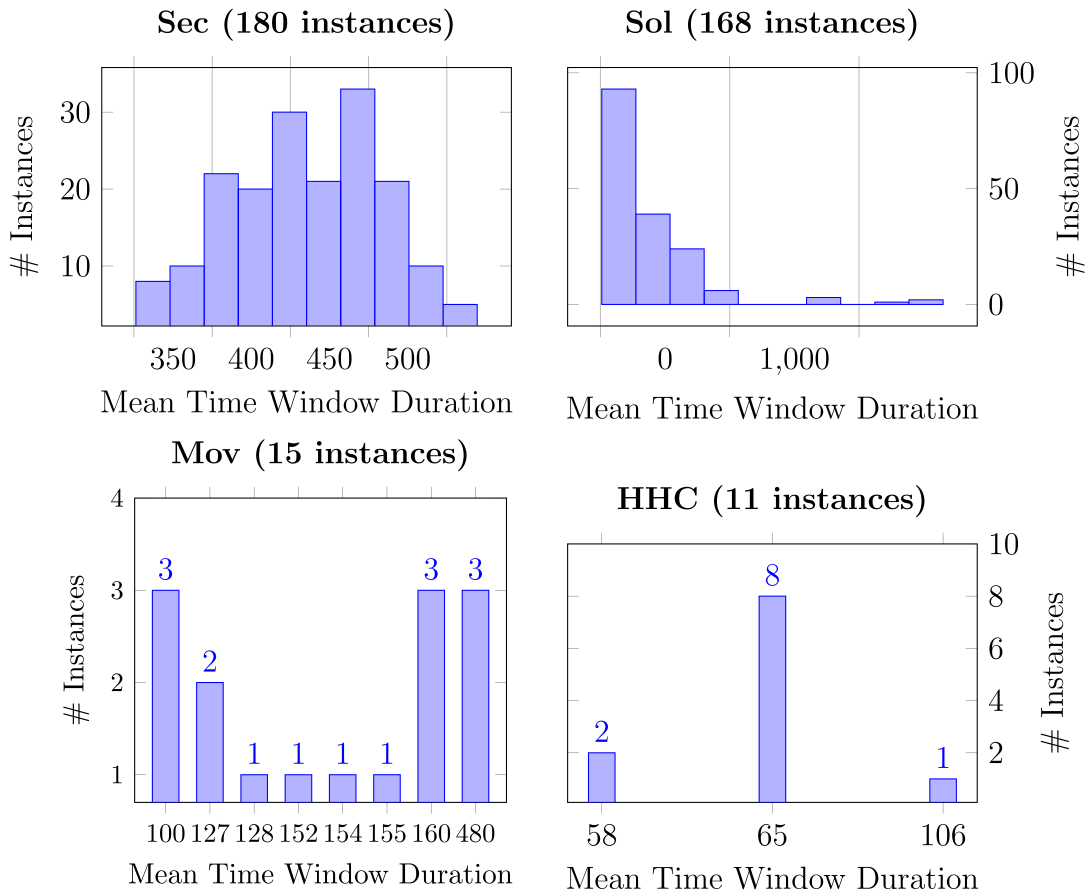

Figure 4.4: Shows the distribution of instances according to their time window duration for every data set (Sec, Sol, Mov, HHC)

4.4.5 Service Time

The service time, i.e. duration of activities, determines the number of effective working hours employees have to provide. If activities are short, then it is expected that the proportion of the travel time increases because it means employees spend more time travelling than providing continuous effective work. One activity of two hours in some cases is preferred to two activities of one hour with travelling required. Activities' duration varies across sectors. In many cases, services are between 15 minutes to one hour maximum. In other cases, activities could be as long as the entire employee's shift. Table 4.6 contains information regarding the average activities' duration for each data set. Sol data set presents no variance between the minimum and maximum which implies that most activities have the same duration in the instances. Mov mean duration of activities is just under 22 minutes. HHC presents a mean activities' duration as short as five minutes up to almost 90 minutes. In contrast, Sec the minimum average duration of activities is around one hour, and the maximum average duration is 14.5 hours. In the latter, an employee assigned such a prolonged activity will only perform this activity. Activities with long duration, similar to the planning horizon, could be assigned separately, and not in the same process of activities that require routing involved. It is only within the Sec instances that this characteristic is encountered.

Figure 4.5 shows the distribution of average time duration of activities per instances in all data sets. Similar to other previous characteristics Sec presents more diversity. The remaining three (Sol, Mov and HHC) have clear distinction in the average values.

#Instances	Data Set	Min	Mean	Max
168	Sol	34.28	34.28	34.28
15	Mov	10.00	21.16	30.00
11	HHC	5.45	26.27	87.27
180	Sec	62.41	409.83	864.58
1	Tech	10.00	150.34	417.00

Table 4.6: Contains average activities' duration within each data set. Time is given in minutes

4.4.6 Planning horizon

The size of the planning horizon is an important aspect of the WSRP as it restricts the number of activities that can be performed and often determines the maximum employee working time. In all instances employees are available during the whole

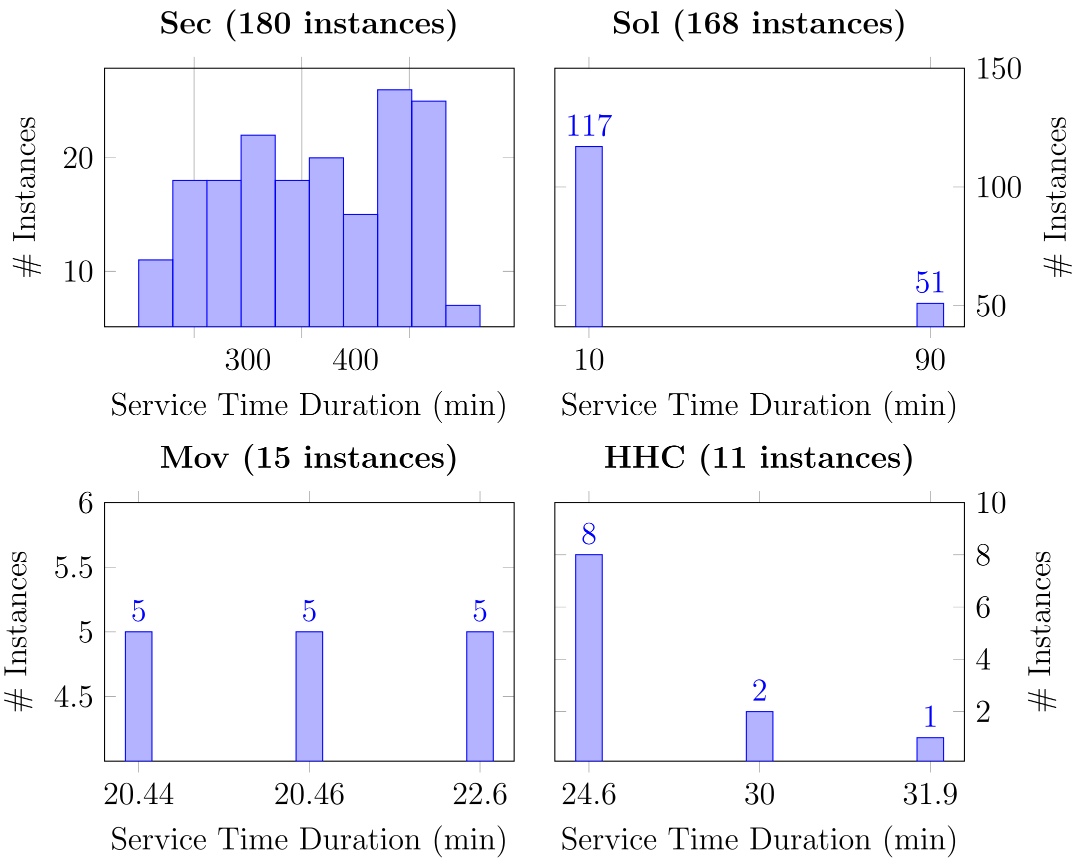

Figure 4.5: Shows the distribution of mean activities' duration within the instances of each data set (Sec, Sol, Mov, HHC)

planning horizon, i.e. their working time is equivalent to the planning horizon. Table 4.7 shows that for the majority of instances the average planning horizon is less than 24 hours (1440 minutes). Only Sol contains some instances above of more than 24 hours with a 2.3 days planning horizon. Clearly shown in Figure 4.6, all instances in Mov and HHC have the same planning horizon: eight hours for Mov and 23 hours for HHC. It is interesting that Sec almost has all instances with the same planning horizon (1440 minutes). There are a few exceptions where the planning horizon is of 25 hours (1500 minutes), this seemed odd as the division of the original monthly district-instances was made on a 24 hour basis. But after further investigation, the exception are days of 25 hours, which can only happen when the date coincides with a day in which daylight savings were adjusted by moving the clocks backward and effectively gaining an hour. Taking that into consideration only Sol present a range of different planning horizon going from four hours up to 2.3 days.

#Instances	Data Set	Min	Mean	Max	Std. Deviation
168	Sol	230	1100.07	3390	1015.32
15	Mov	480	480.00	480	0.00
11	HHC	1380	1380.00	1380	0.00
180	Sec	1440	1442.00	1500	10.80
1	Tech	1380	1380.00	1380	-

Table 4.7: Distribution of planning horizon duration within each data set. Time is given in minutes

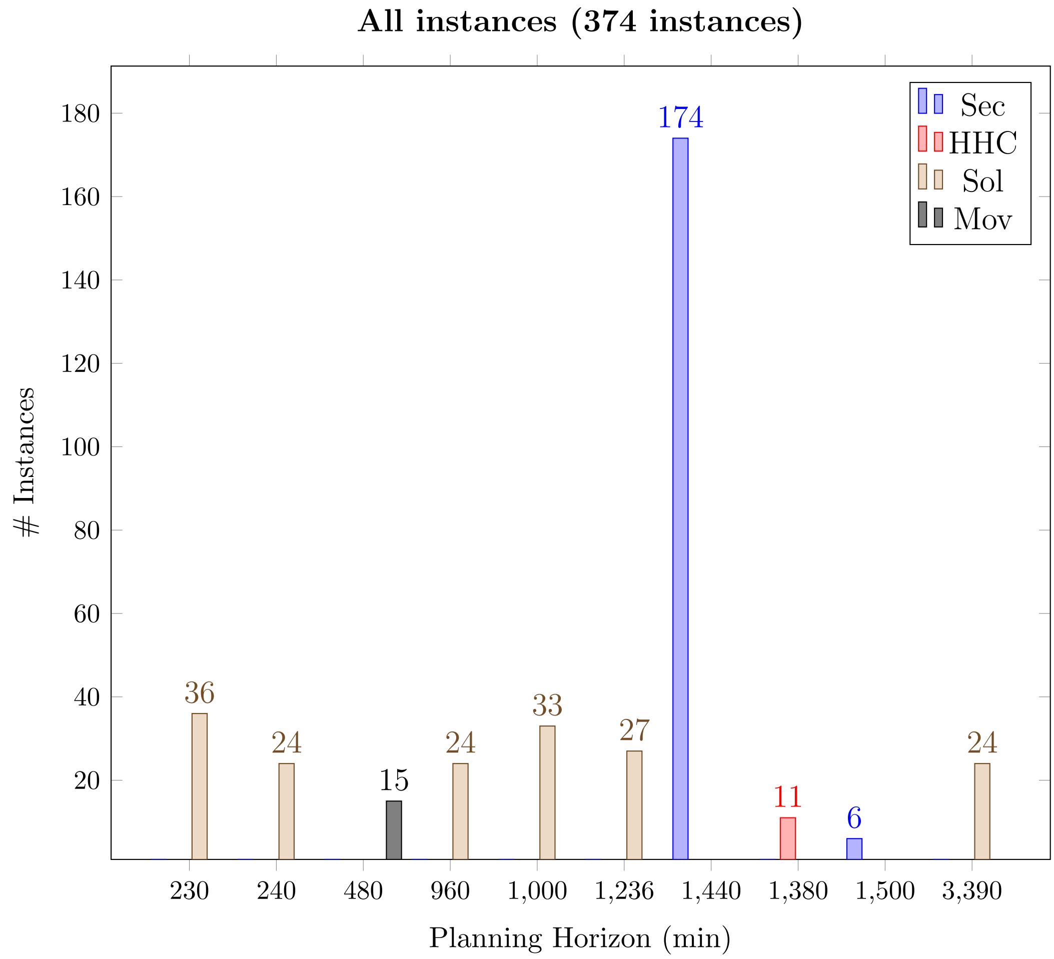

Figure 4.6: Shows the distribution of planning horizon duration across all data sets (Sec, Sol, Mov, HHC)

4.4.6.1 Ratio working capability/work demand

An estimate of the number of hours required to complete all activities could be obtained by multiplying the number of activities by their duration. Such a value provides the effective hours, i.e. time working on activities. The real working time of the instance is only known when all travelling times are known. The available working time of the workforce is estimated as the number of employees multiplied by the percentage of activities that the average employee can perform based on skills multiply by his shift time (in this case planning horizon). Table 4.8 provides the ratio between the established working capability of the workforce and the work demand for every data set. Sol and HHC have a ratio of about 4.5 times available working capability to perform all the activities required. Mov has even more, eleven times more working capability than required. Although just an estimate, it was previously confirmed that the Sec data set was over staffed, and a ratio of 11.38 confirms it. In contrast, the ratio of Sec is below 1.0 which means that for some of these instances not all activities can be performed because there are not enough qualified employees available to cover all activities.

#	Data Set	$\mu_{Activities} * \mu_{Duration}$	$\mu_{Employees} * \mu_{skill} * \mu_{TimeHorizon}$	Ratio
168	Sol	2000.00	9197.81	4.59
15	Mov	3175.33	36150.40	11.38
11	HHC	2677.66	12493.51	4.66
180	Sec	44280.44	33693.81	0.76

Table 4.8: Ratios between the established working capacity of the workforce and the work demand

The previous ratio is just an estimate, other factors such as the number of synchronised activities or the number of employees required, need to be considered in order to increase confidence in whether all activities in an instance can be performed. Both considerations determine that a number of employees must perform an activity at the same time, so if an activity requires a team of four, but there are only three employees in the workforce then such activity will not be performed regardless of the instance having spare working time at other time periods.

4.4.7 Teaming and time-dependent activities constraints

These two features vary across the data set. They are considered together because a teaming requirement can be modelled using synchronised constraints (time-dependent of type synchronisation). For example, if an activity requires a team of three employees, that can be modelled with two synchronised constraints. Table 4.9 presents a summary on the presence of teams and time-dependent activities constraints within the data sets. Synchronisation and overlapping occur in the majority of instances. Min-max type is found only in Mov and Sec. Apart from HHC, the rest of the data sets contain instances with activities that required two employees to be performed.

Figure 4.8 shows the distribution on the number of time-dependent activities con-

#	Data Set	μ_{Sync}	μ_{Over}	μ_{Min}	μ_{Max}	$\mu_{Min-max}$	$\mu_{Team(2)}$
168	Sol	2.66	2.99	0.66	2.00	0.00	6.00
15	Mov	5.40	5.53	4.00	3.33	3.33	14.66
11	HHC	0.90	0.18	0.00	0.00	0.00	0.00
180	Sec	3.31	3.24	4.30	4.07	4.41	21.89

Table 4.9: Average number of time-dependent activities constraints and number of employees required per activity across all data sets (Sol, Mov, HHC, Sec)

straints. There is one graph for each of the following types of constraints: synchronisation (Sync), overlapping (Over), minimum difference (Min), maximum difference (Max) and minimum-maximum difference (MinMax) for every data set Sec, Mov, Sol and HHC. Not every type of constraint is present in each data set, e.g. Min-HHC, Max-HHC, MinMax-HHC, MinMax-Sol. Moreover, Tech does not contain any time-dependent constraints; hence no graph is required for this data set.

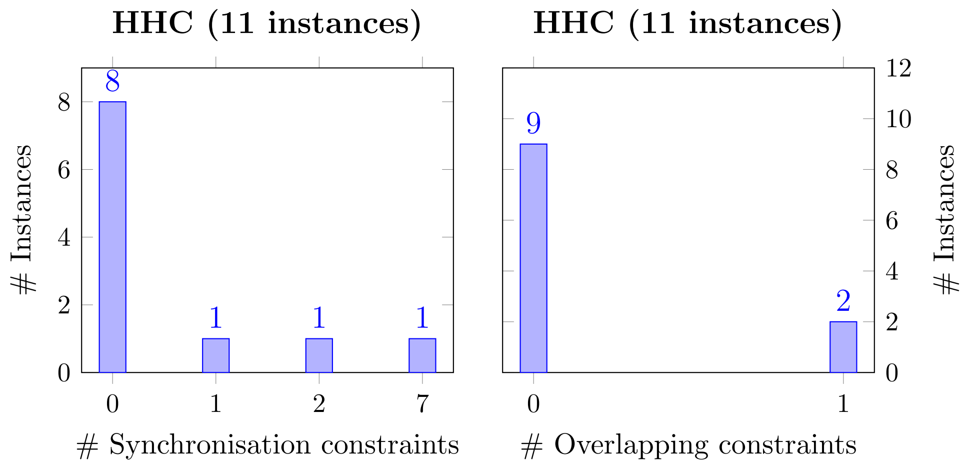

Figure 4.7: Showing the distribution of time-dependent activity constraints of type synchronisation and overlap for the HHC data set.

4.5 Summary

In this chapter the data sets used in this PhD thesis were presented. Details regarding the original source of each data set were provided. In addition, the additional features and modifications performed were also explained. Finally, analyses of: number of activities, employees, skills, time windows, duration of activities, planning horizon, teaming and time-dependent activities constraints was provided for each of data set. In the following chapters, the adapted instances are used to experiment with the models and algorithms presented in this thesis. When appropriate, references to the

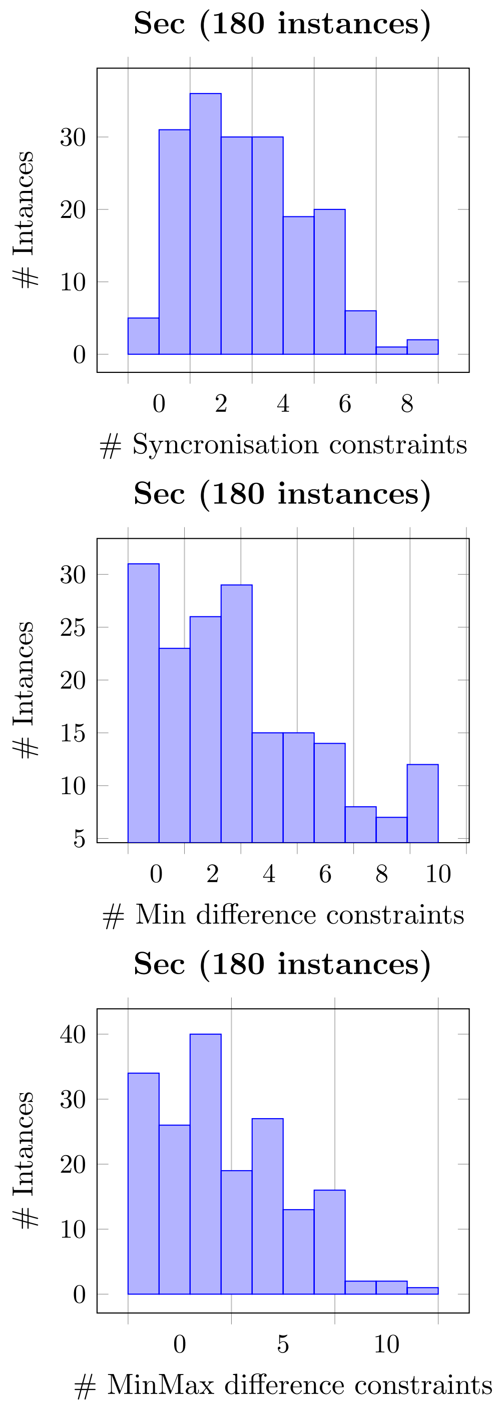

Figure 4.8: Showing the distribution of time-dependent activity constraints of type synchronisation, overlap, minimum, maximum and min-max difference for Sec.

analysis performed in this chapter are used to convey research findings.

Table 4.10 provides a summary of the characteristics added, removed or changed for each data set as a reference.

In Appendix A, Table A.1 contains detail information regarding number of activities,

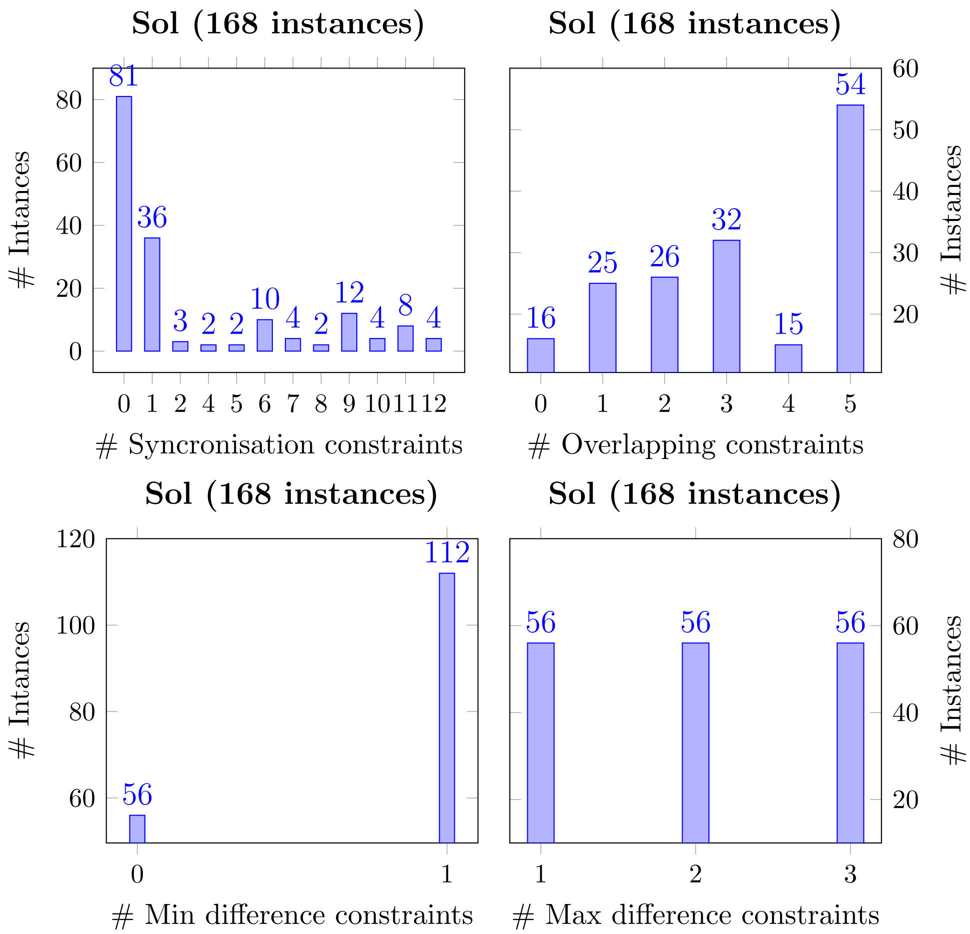

Figure 4.9: Showing the distribution of time-dependent activity constraints of type synchronisation, overlap, minimum and maximum difference for the Sol data set.

number of employees, employees's coverage of activities based on skills, mean time window duration, mean service time, planning horizon duration and number of time-dependent constraints for each instances. The table is the base data from which the summary information presented in this chapter was obtained.

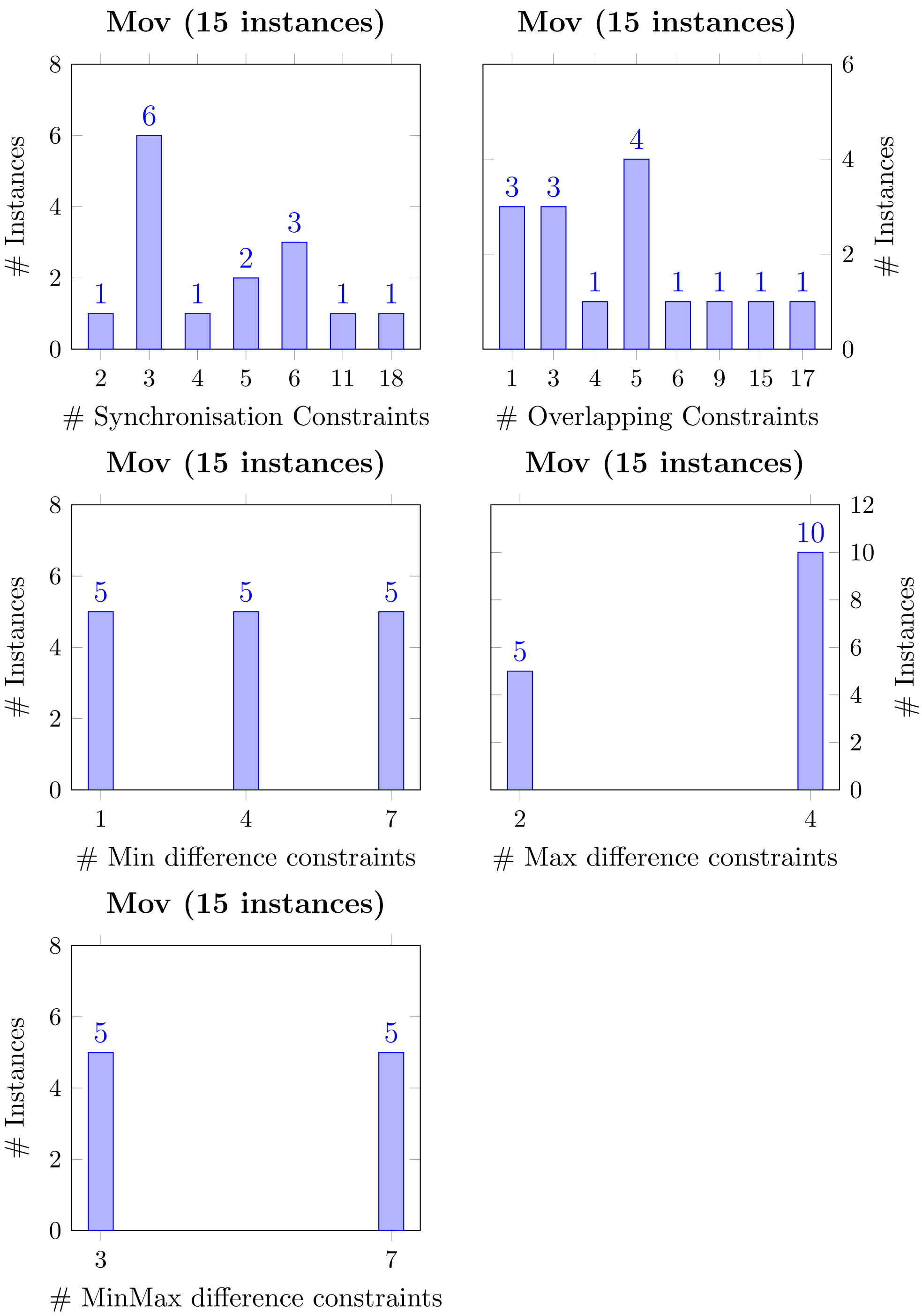

Figure 4.10: Showing the distribution of time-dependent activity constraints of type synchronisation, overlap, minimum, maximum and min-max difference for Mov.

Characteristic	Sol	Mov	HHC	Sec	Tech
Number of employees	A	-	-	C	-
Additional instances	A	-	-	C	-
Skills definition/addition	A	A	-	-	-
Employees working time	A	-	-	-	-
Activities requiring Teams	A	A	-	A	A
Preferences addition	A	A	-	A	-
Connected activities constraints	A	A	-	A	A
Time horizon definition	-	-	-	A	-

Table 4.10: Summary of characteristics that were added (A) or changed (C) from the original data set.

Chapter 5

Mathematical Programming Models

5.1 Introduction

This chapter adapts two mathematical models from the literature. The first is an Integer Linear Programming Model (IP) used in the Vehicle Routing Problem with Time Windows (VRPTW). The second is a Mixed Integer Linear Programming Model (MIP) which among other features allows activities to be left unassigned. It will be clear that the MIP expands on the features of the IP. The chapter is divided in two sections, each of them covering one of the two models.

5.2 IP Model

In the IP model all activities in the instances should be feasible scheduled and performed in order to obtain a feasible result, i.e. a schedule indicating which employees are performing which subset of activities indicating their sequence and starting time. This approach may present problems, highlighted during the analysis of the entire data set in the previous chapter, in that some instances do not have enough employees to cover all activities. Nonetheless, for other instances it should be possible to assign all activities, since it appears there is enough working time available to perform them.

This section covers the following three objectives. The first one is to use Bredström and Rönnqvist (2008) VRPTW model to tackle WSRP by performing the necessary modifications to include all additional constraints. The second objective is to assess if WSRP problems are more difficult in comparison to VRPTW ones. A comparison is useful in order to ascertain whether WSRP requires less computational effort to solve than VRPTW. If this is case, then current approaches to tackle VRPTW should suffice. If WSRP is harder to solve than VRPTW, the need for new/adapted models and algorithms is justified. The third objective is to discuss the results of the mathematical solver (Gurobi) when tackling WSRPs using the IP Model.

5.2.1 IP Model Description

Given the requirement of assigning all activities, the Bredström and Rönnqvist (2008) model presents the following advantages compared to other models available in the literature. The Korsah et al. (2010) model includes waiting times in the definition. Waiting times are the idle periods in which employees are neither performing activities nor travelling, e.g. when arriving to a location before the time window opening employees are required to wait until it occurs. Such approach greatly increases the number of variables that are generated for the model. In smaller instances, i.e. with less than 25 activities and an equal or smaller number of employees, this is not an important issue as the solver can handle it, but one observed difficulty reported in the literature for VRPTW when using mathematical solvers refers to the amount of memory being used for big models, i.e. more than 100 activities. Knowing waiting times, although desirable as a performance indicator, it is not necessary when the aim is to cover all activities. Another model by Rasmussen et al. (2012) considers that in some cases assigning all activities is not possible and therefore introduces additional variables to allow the possibility of unassigned activities. Their approach is discussed in the next section (see Section 5.3). The IP model is as follows:

$$\min \qquad \alpha_p \sum_{k \in K} \sum_{(i,j) \in A} c_{ik} x_{ijk} + \alpha_T \sum_{k \in K} \sum_{(i,j) \in A} T_{ij} x_{ijk} \tag{5.1}$$

$$s.t. \quad \sum_{k \in K} \sum_{j:(i,j) \in A} x_{ijk} = 1 \quad \forall\, i \in N, \tag{5.2}$$

$$\sum_{j:(o,j) \in A} x_{ojk} = \sum_{j:(j,d) \in A} x_{jdk} = 1 \quad \forall\, k \in K, \tag{5.3}$$

$$\sum_{j:(i,j) \in A} x_{ijk} - \sum_{j:(j,i) \in A} x_{jik} = 0 \quad \forall\, i \in N, \quad \forall\, k \in K, \tag{5.4}$$

$$t_{ik} + (Tij + D_i)x_{ijk} \leq t_{jk} + b_i(1 - x_{ijk}) \quad \forall\, k \in K, \quad \forall\, (i,j) \in A, \tag{5.5}$$

$$a_i \sum_{j:(i,j) \in A} x_{ijk} \leq t_{ik} \leq b_i \sum_{j:(i,j) \in A} x_{ijk} \quad \forall\, k \in K, \quad \forall\, i \in N, \tag{5.6}$$

$$a_i^k \leq t_{ik} \leq b_i^k \quad \forall\, k \in K, \quad \forall\, i \in o, d, \tag{5.7}$$

$$\sum_{k \in K} t_{ik} = \sum_{k \in K} t_{jk} \quad \forall\, (i,j) \in P^{sync}, \tag{5.8}$$

$$\sum_{k \in K} t_{ik} + p_{ij} \leq \sum_{k \in K} t_{jk} \quad \forall\, (i,j) \in P^{prec}, \tag{5.9}$$

$$x_{ijk} \in \{0,1\} \quad \forall\, k \in K, \quad \forall\, (i,j) \in A, \tag{5.10}$$

$$t_{ik} \in \mathbb{Z}_+ \quad \forall\, k \in K, \quad \forall\, i \in N. \tag{5.11}$$

In this model, N is the set of activities' locations. The node o refers to the starting point of the employees. Node d denotes the final destination of employees after completing their activities. In this model nodes (o and d) represent the same node if the starting and ending location is the same, but still two nodes are required due to the nature of the model, i.e. based on network flow. Set A contains all the locations in N plus the two extra locations for starting and ending nodes. If the start and end location are different for every employee then employees' starting and ending locations are also included in A. The set of all available employees is represented by K. Every activity i defines a time window on its starting time. The time window is given by a_i (earliest start time) and b_i (latest start time). Activity i's duration is given by D_i. Travel time between location i and j is considered in the integer variable T_{ij}. Variable t_{ik} is a binary variable that indicates whether employee k performs the activity at location i. Note, if two or more activities have the same location but cannot be performed on the same visit then additional variables for every activity are required due to the nature of the model. Employee k's working time is given by a_i^k (start time) and b_j^k (end time). The constant E_{ij} considers the travelling time between locations i and j plus the duration of activity at i, i.e. $E_{ij} = T_{ij} + D_i$. Using such a constant assumes that as soon as an employee finishes an activity he starts travelling for his next assignment immediately. If the employee arrives at the next location early, before the opening of the time window, he has to wait.

The objective function (5.1) has two components. The first component is the cost of assigning activities to employees. Such cost is given by c_{ik} for activity i performed by employee k. The second component is the travel time of all employees when performing their visits. Both components have a weight factor. The cost component's weight is α_p, and for travel time component is α_T. Such weights can be set accordingly depending on the units being used or the importance given to any of the components. Using weighted sums as objective function has the advantage of allowing more than one aspect of the WSRP to be considered e.g. employees' costs and travel time. Weights can be adjusted depending on which component has more relevance for a given scenario. A disadvantage of weighted sums is the loss of a common unit of measure i.e. money and time. In addition, sometimes a negative value in a component's sum could diminish the result of another component. In multi-objective optimisation weighted sums are often used. Nevertheless, it has disadvantages as sometimes it fails to locate Pareto optimal solutions (Ward Athan and Papalambros, 1996).

The constraints are described as follows. All visits must be performed (constraint 5.2). All employees must start (leave) from location o and return to location d (constraint 5.3). Constraint (5.4) maintains flow conservation, i.e. once an employee visits a location represented by a node, to perform an activity, he must leave the location. Constraint (5.5) ensures that the integer variable capturing the start time of activity i is less than the next activity j which the employee performs, avoiding cycles. Each visit's time window must be met, constraint (5.6) enforces the start time to be within the time window (a_i^k–b_j^k). Visits should be performed during employee's working time (5.7). Synchronisation constraints (5.8) are necessary for every pair of visits that need to be synchronised. Other types of time-dependent activities constraint are enforced in (5.9). Decision variable $x_{ijk} = 1$ when employee k travels from location i to j, it assumes it performs the activity at i. Or, $x_{ijk} = 0$ if the employees does not use that segment of the graph, i.e. does not perform activity i and travel towards j. Constraint (5.10) restricts such variables to be binary. Variables recording the starting time of activities are positive integers (constraint 5.11).

5.2.2 Modifications to IP model

There are some modifications to the Bredström and Rönnqvist (2008) model. The first one considers using positive integers instead of real variables to record the starting time of activities (5.11). Such changes allow the representation of the starting time in minutes or seconds depending on the accuracy required. In some sectors, recording to the nearest 15-minute period is enough. As shown in the analysis of the instances (chapter 4) the majority of the instances have a planning horizon of less than 24 hours. Using an accuracy in minutes gives 1440 possible values for such variables (t_{ik} represents a given minute in the planning period).

Another modification to the original model is in the objective function (5.1), the removal of a balancing component and its associated weight (see Bredström and Rönnqvist, 2008, pg. 25). Such a balancing component could be included when factors like fairness on service time or workload for every employee are taken into account. Fairness on assignments of activities has different meanings depending on the sector. It might be a balance assignment of working time, or the same number of visits per employee, or each employee performing their preferred visits to a certain degree etc. Within the instances such notion is not included or mentioned. As a result, the balancing component was removed, leaving the objective function only with cost and travel time.

5.2.3 Experiments using the IP Model

The aim of the experiments is to obtain optimal solutions if possible. Since the IP model is based on the VRPTW problem, only the data sets based on such problem are used (Sol and Mov). Another reason for using only those data sets, is that the model requires to complete all activities in the instances. From the analysis performed on the Sec data set, it is concluded that in many instances there are not enough employees available to perform all activities, as consequence the Sec data set was discarded in the experiments in this section.

The weights used in the objective function α_p and α_T are given both the same value, in this case (0.5) since no particular additional component is more important than the other one.

The experiments were carried out using Gurobi solver version 5.1 and executed on a X64-based PC running Microsoft Windows 7 Enterprise with 2 Duo CPU (3.16 GHz) and four gigabytes of RAM.

5.2.3.1 Tackling the first objective

The objective of tackling WSRP with a model based on VRPTW is investigated by performing the following experiments. Both data sets Sol and Mov are solved by a commercial solver (Gurobi) in order to obtain optimal, and if not possible, at least good integer feasible solution. No time limit was set for the solver, therefore the optimisation process stops when finding the optimum value, when the solver runs out of memory, or when the solver proves the instance being integer infeasible.

5.2.3.2 Overview of Results

The experiments used in total 183 instances (Sol and Mov). The solver was not able to load instances of 150 and 250 activities due to the size of the model (the amount of memory required), such instances belong to the Mov data set and in total are 10, five of each number of visits. For the remaining ones 173, Table 5.1 shows the overall results. There were four types of results among the 173 instances: instances for which the solver found optimal solutions (Optimal), instances for which the solver reported as infeasible (Infeasible), instances where the solver run out of memory (OoM) without giving any result, and instances where the solver reported errors (Error).

From Table 5.1 we observed that optimal solutions were only reported in two instances both with 25 activities. It took 67 hours for the longest one to find the optimal solution and 21 hours for the shortest one. Almost half of the instances of 25 and 50 visits run out of memory when searching for the optimal solution, nevertheless feasible solutions were found. The time by which the solver reports the first available solution is provided in parentheses. It is noticeable that the first feasible solution is reported as fast as eight seconds and maximum of four hours, mean computation time is between 10.5 (for 25 visits instances) and 5365 (for 50 visits instances) seconds. Infeasible solutions could be identified almost immediately, in less than one second after the optimisation commences, but for other cases after 45 hours.

The overall observation is that these WSRP instances are computationally difficult to solve for the mathematical solver with this IP formulation. More importantly, it is observed that for a good proportion of them no feasible solution was obtained. For example, none of the 100 activities instances was solved. It is clear that for instances of that size the solver using this model is not a good option as solution method.

Figure 5.1 shows the gap reduction of an optimal instance found by the mathematical solver. The outer graph represents all time required to achieve optimality (241620 seconds). The inner graph only shows the first two hours (7200 seconds) which in this case is the amount of time required for the solver to achieve almost a 10% gap. Reducing the gap further takes the solver 65 additional hours. Hence, setting a maximum time limit of two hours for further experiments is a reasonable compromise. It should be remembered that the instances represent daily planning problems where two hours waiting for a result from the solver is a significant but manageable time.

Total	Size	Outcome	# Instances	Min(t)	Mean(t)	Max(t)	Std Dev(t)
56	25	Infeasible	30[a]	0	5417.80	162340	29637.88
		Optimal	2[a]	79072	160346.00	241620	114938.79
				(8)	(10.50)	(13)	(3.53)
		OoM	21[a]	38430	117198.48	348129	75519.83
				(188)	(2329.08)	(9293)	(2599.13)
		Error	3[a]	-	-	-	-
61	50	Infeasible	31[a]	0	24.29	331	73.49
		Optimal	0[a]	-	-	-	-
		OoM	26[a,b]	1651	67294.31	269150	61362.68
				(63)	(5365.00)	(14898)	(8273.17)
		Error	4[a]	-	-	-	-
56	100	Infeasible	53[a]	0	55.64	123	32.27
		Optimal	0[a]	-	-	-	-
		OoM	0[a]	-	-	-	-
		Error	3[a]	-	-	-	-

Table 5.1: Summary of the outcome infeasible, optimal, out of memory (OoM) reported by Gurobi. Computation times t (Minimum, Mean, Maximum, Standard Deviation) are in seconds. Times in parenthesis show when the solver found the first feasible solution when available. [a] Sol data set. [b] Mov data set.

In the next round of experiments we concentrate in the 25 and 50 visits size instances.

5.2.3.3 Tackling the second objective

Two of the constraints, 5.8 and 5.9, are included to tackle the case when activities require more than one employee and when activities have time-dependent relationships. It could be argued that the presence of these two additional type of constraints could make the search easier since the search space is reduced. Or, it might be the case that introducing the constraints makes the problem computationally harder to search because, although the search space might be reduced, it might also be divided in small separated feasible areas, and finding those areas could be more difficult. A way of testing whether Teaming and time-dependent activities constraints make the search easier or more difficult for the solver is to use the same model with some instances that include those constraints and other instances that do not include them and compare results.

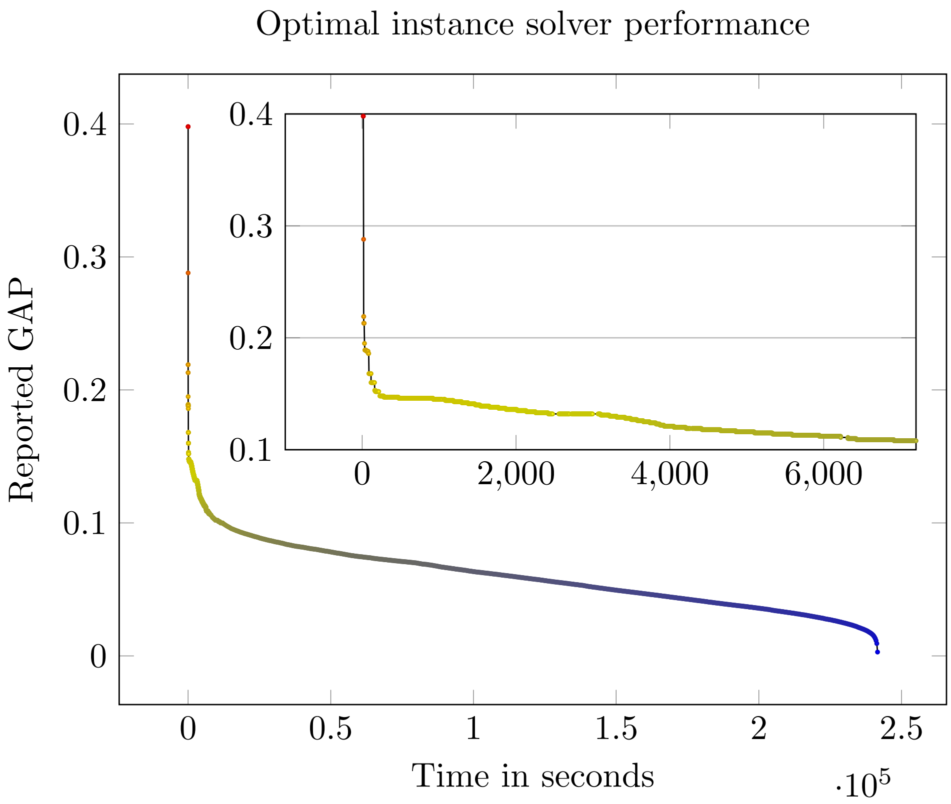

Figure 5.1: Gap reduction as computation time progresses in a case in which the mathematical solver found the optimal solution. The optimal solution is reported after 241620 seconds (approximately 67 hours) but a considerable gap reduction is achieved during the first two hours, as shown in the close up

5.2.3.4 Experiment Design

The experiments use only the instances of activities size of 25 and 50 within the Sol (112 instances) data set. Two runs of experiments are performed. The first run has no changes of the instances. In the second run, all teaming and time-dependent activities constraints are removed from the instances. For both runs of experiments the same IP model, computer and mathematical solver settings are used. The time limit set was 15 minutes. This time limit differs from the suggested two hours from the first objective as the purpose is to see the effect of having time-dependent activities or not and not to find the optimum value. The same objective function is used (5.1). At the end of both experiment runs a comparison is made to identify which group of instances the solver achieves better results. Such comparison might provide guidance in answering whether teaming and time-dependent activities constraints make the search more difficult or not.

5.2.3.5 Experimental Results

To facilitate the report of experiments we divided the instances according to the original Solomon groups, i.e. C100, C200, R100, R200, RC100, RC200. Table 5.2 shows the number of instances in each group for which the solver found a feasible solution. The solver found more feasible solutions in all groups where the version was without Teaming and Time-dependent Activities Constraints (TTC). In total there were 59 feasible solutions compared with 26 for the version with TTC constraints.

TTC Constraints	C100 (18)	C200 (16)	R100 (24)	R200 (22)	RC100 (16)	RC200 (16)	Total (112)
With TTC	10	8	0	4	0	4	26
Without TTCs	13	13	3	12	8	10	59

Table 5.2: Number of feasible solutions found in every group. Values in parentheses indicate the number of instances per group. Teaming and time-dependent activities constraints (TTC)

The version without the teaming and time-dependent constraints finds more feasible solutions (59) compared to the version that includes them (26). Although better results are obtained the version without constraints still remains a difficult problem for the solver since just 59 out of 112 instances could be solved. Table B.1 (see Appendix B section B.1) provides details of the results of each instance with TTC constraints. The results include best objective, lower bound found, and the respective gap. It also shows whether the solver identifies the instance as infeasible. The solver reported overall: 26 instances with feasible solutions, out of which seven were optimal ones; seven instances were infeasible; and, for the remaining 79 instances, the solver time out without reporting any feasible solution. Similarly, Table B.2 (see Appendix B section B.2) provides detail of individual instances for the version without TTC constraints. The solver found feasible solutions for 59 instances, out of which 15 were optimal. There were two infeasible instances. And, for the remaining 51 instances the solver timed out without reporting any feasible solutions. There are 26 instances that have feasible values for both experiments (with and without TTC), the solver obtains the same results for seven of them. These seven are the optimal ones reported with TTC. For the rest, the version without TTC achieves a better gap. A series of Figures showing the reduction in gap as computation time progresses for both experiments runs is included in Appendix B for results with TTC refer to B.1, B.2, B.3, B.4 for results without TTC refer to B.5, B.6, B.7, B.8, B.9, B.10.

Regarding infeasible results in both experiments, two instances are infeasible due to the lack of employees to cover all activities, whether because there are not enough or they do not have the required skills. The remaining five infeasible instances are due to conflicts when introducing TTC, particularly those constraints that require the simultaneous performance of two different activities, because they require two or more employees available at a specific time.

5.2.3.6 Review of the First objective: Varying time limit

In the experiments of section 5.2.3.2 there was no limit in computation time, in the hope that the solver could eventually find optimal solutions. In this section, three additional time limits are used for instances with 25 and 50 visits. The time limits are 15 minutes, 60 minutes and 240 minutes. The need for repeating the experiments with an specific time limit was to obtain more information regarding instances that run out of memory in the previous section. 5.2.3.2.

Results from section 5.2.3.5 for instances with TTC with time limit of 15 minutes are re-used here. There are 79 instances for which the solver timed out after 15 minutes providing no result, for those instances only the time limit is increased to an hour. Table B.3 (see Appendix B section B.3) shows detailed results of individual instances. The solver only found three instances with new feasible solutions. Figure B.11 (Appendix B) shows the gap reduction of the three instances, notice how the x-axis starts around 2000 seconds.

There are still 76 instances for which the solver does not provide any information apart from the lower bound. A third increase in time limit is performed (240 minutes). The number of minutes is chosen to maintain the same ratio (four times) as for the previous two set of experiments (15 to 60) and (60 to 240). Table B.4 (see Appendix B section B.4) contains detailed information for each instance. The solver found 16 new feasible solutions. The new results belong to all groups except R100. Figures B.12, B.13 and B.14 also in Appendix B illustrate the gap reduction for the 16 instances. Among all groups R100 contains instances with shorter time horizon with respect to the other groups. The duration of the time horizon is equivalent to employees working time. In other words, instances in R100 have the same work with less resources (employees-hours). For the same reason six out of seven infeasible instances belong to that group R100.

5.2.3.7 Tackling the Third objective: Gurobi results

Gurobi provides information regarding the current gap achieved while performing the optimisation. In the experiments, Gurobi is set up to report the gap reduction every 15 seconds. When a gap reduction is achieved, the method used is reported by the solver. The objective in this set of experiments is to identify which method is used by Gurobi when finding better solutions for each instance. For every new feasible solution Gurobi reports whether the solution was found by branching or by heuristics as specified in the reference manual of the solver Inc. (2013). If most of the time new feasible solutions are found by heuristics, it would justify developing a tailored one for WSRP. In all previous experiments, without exemption Gurobi found more gap improvements when using heuristics. It is expected that MIP heuristics find more feasible solutions than the branching process for the VRPTW. The adaptations to the data set and modification of the VRPTW model to tackle WSRP have similar results. In fact, the number of times a heuristic within Gurobi finds a better solution is in general larger for instances that include the additional constraints in the WSRP instances.

Table 5.3 summarises the number of times a gap reduction was achieved for every group of instances in all experiments. The table has four rows but split in two parts vertically, each part has three groups of instances. Note that the second row in each part, marked with (*), refers to all instances without the teaming and time-dependent activities constraints. The third row in each part shows the 79 instances that timed out after 15 minutes in the first set of experiments but then executed for up to 60 minutes. The number in parentheses after the time limit is the number of instances used in that set of experiments. In all groups there are more gap reductions achieved by heuristics than by branching (H/B values).

5.2.4 IP Model Remarks

The computational experiments performed in this section provide solid evidence that WSRP instances are more challenging to solve using the IP model and mathematical solver under the described conditions than their VRPTW counterpart, from which they were generated. Table 5.4 summarizes the results of this section. The generated WSRP instances are more difficult to solve due to the additional teaming and time-dependent activities constraints (similar results are reported by Rasmussen et al. (2012)). Additionally, it is found that WSRP instances with clustered visiting locations tend to be easier to solve according to the gap percentage reported by the solver

Time(#)	C100	H/B	C200	H/B	R100	H/B
15m(112)	10	26/16	8	27/11	0	0/0
*15m(112)	13	50/12	13	71/19	3	10/5
60m(79)	0	-/-	0	-/-	0	-/-
240m(112)	14	95/23	10	83/23	0	0/0
Time(#)	R200	H/B	RC100	H/B	RC200	H/B
15m(112)	4	25/10	0	0/0	4	57/14
*15m(112)	12	75/21	8	48/25	10	99/20
60m(79)	2	8/1	1	3/2	0	-/-
240m(112)	11	196/27	2	11/1	8	126/15

Table 5.3: Summary of methods used by Gurobi during the optimisation process. Columns H/B report the number of gap reductions within a group of instances that are achieved with Heuristics (H) or Branching (B). Within every group the number of instances with feasible solutions is reported.

in the experiments.

The computation time limit for a mathematical solver to find good feasible solutions for the generated WSRP instances in data sets (Sol and Mov) needs to be more than one hour. Considering only the 45 instances for which feasible solutions are found, the solver took less than an hour for 29 instances. For the remaining 16, feasible solutions are found within one to four hours. Nevertheless, for 90% of 45 instances, feasible solutions are found within two hours and five minutes. Adding two more hours of computational time achieved only 10% more feasible solutions. This is not practical, hence it is suggested to use a maximum computation time of two hours when solving WSRP instances with planning horizon of one day.

TTC	Time	C100 (18)	C200 (16)	R100 (24)	R200 (22)	RC100 (16)	RC200 (16)	Total (112)
Without TTC	15 min	13(8)	13(4)	3[2]	12(1)	8(1)	10(1)	59(15)[2]
With TTC	15 min	10(5)	8(2)	0[6]	4	0[1]	4	26(7)[7]
With TTC	60 min	10(5)	8(2)	0[6]	6	1[1]	4	29(7)[7]
With TTC	240 min	14(5)	10(2)	0[6]	11	2[1]	8	45(7)[7]
Unknown		4	6	18	11	13	8	60

Table 5.4: Summary of experiments. Number of instances for which the solver achieves feasible solutions. Values in parentheses () in the header row refer to the total of instances in the group. Values in parentheses () within the data indicate the number of optimal solutions. Values in brackets [] report the number of infeasible solutions within the group

5.3 MIP Model

In this section, the IP Model (see section 5.2) is changed to one that allows activities to be left uncovered (unassigned). It is clear from the results obtained in the experiments that if the constraint that forces all activities to be assigned is not relaxed (see constraint 5.2), the solver would not be able to provide feasible results for some scenarios, i.e. understaffed ones. The solver on the IP model does not provide information on the number of activities that could not be performed as the result is the same whether one activity or hundreds are left unassigned, i.e. an infeasible solution.

The scheduling variables t_{ik} are changed to rational values rather than integer ones as this action could benefit the solver. Fewer integer variables reduce the memory requirements of the branch and bound tree. The MIP model introduces another set of binary decision variables for every activity per employee. This model, although requiring more memory resources when tackled by a mathematical solver, provides feasible solutions for the majority of instances as a feasible solution now can include unassigned activities unlike in the IP version. There are some instances that remain unsolvable due to their size with the hardware configuration used, i.e. the solver cannot load them. Nonetheless, the solver provides lower bounds and reports on the gap between such lower bounds and the best feasible solution found for each instance. A method for reducing the number of variables in models that include Teaming requirements is discussed. The experiments with the MIP model use all five data sets (Sol, Mov, Sec, HHC and Tech) for a total of 375 instances. The difference in the experiment settings is that the solver is now allowed to run for only two hours for each instance, unlike the unlimited execution time that was used before (see section 5.2.3.1). Finally, results are provided using a new objective function. The change in the objective function was necessary to include penalties for unassigned activities. In addition, the new objective function could also consider other characteristics, that became apparent at a later stage during the research programme, such as employees' preferences and activities' priorities. The results of this chapter establish a benchmark for the remaining solution approaches in the following two chapters 6 and 7.

Services industries such as home health care (HHC)(Bostel et al., 2008) and technicians in the field (Doerner and Hartl, 2008) present unexpected visits added to a daily plan. These visits are often the result of emergency situations. For example, in HHC, it could be taking an elderly patient to the hospital after reporting signs of a possible hearth attack. Or, in the services sector, it could mean attending a burst water pipe at a customer location so as to prevent water wastage. In the services industry activities are a top priority and are often priced accordingly. That means the customers have the right to solicit an immediate response to their emergency. Addressing these emergency visits is normally addressed in three ways: 1) via the use of casual (agency) employees; 2) through overtime of current members of the workforce; and, 3) allocating low priority activities to another day and fitting in the emergency activities instead. Which approach to choose depends on the circumstances and the industry. Choosing one approach does not preclude using the others. However, if the objective includes cost-reduction, then delaying a low priority task in order to attend an emergency does not have an immediate associated cost compared to the other two approaches, i.e. one day's casual hire or overtime. Casual workers tend to be paid at a much higher rate and overtime in the services industry is often also paid at a premium (Van den Bergh et al., 2013, pg. 370). It is necessary to allow activities to be unassigned in order to support the solution method for which low priority visits should be delayed in order to address emergency activities.

5.3.1 From IP to MIP: model adaptations

The decision to provide a model that allows for activities to be left unassigned is based on two reasons. The first reason relates to instances in data set Sec where there is clearly insufficient employee-hours available to cover all activities (see 0.76 ratio in Table 4.8). Insufficient employee-hours results in an infeasible instance. Moreover, the solver does not provide a partial solution that at least performs as many activities as possible. The second reason is that there are instances neither with feasible solution nor lower bound that are not reported as infeasible. The solver seems to struggle with this type of instances. It is worth investigating the causes of such difficulty. A model that allows for unassigned activities could not result in instances having infeasible solution due to the lack of skilled employees or working hours in general. Using the MIP model, the solver provides feasible solutions almost immediately, as an empty solution, i.e. with no activities assigned, is still regarded as a feasible one, heavily penalised, but feasible. An additional observation from the previous chapter was that the solver was unable to provide a gap for some instances after four hours of computation time. The MIP model helps to distinguish instances reported infeasible due to time related constraints from understaffed instances.

The previous integer variables t_{ik}, that capture the time unit when activities start, can benefit from being modelled as a continuous variables. Relaxing the integer constraints in such variables facilitates the solver's computational effort. In addition, the values in many of the instances' time matrices are given as rational values which often results in precision issues due to rounding when treated as integers. Since time is continuous, it ought to be modelled in the same way. Finally, a pure integer model tends to be harder to tackle for the mathematical solver than if some variables are rational numbers. The mathematical solver uses branch and bound to solve integer models. By reducing the number of integer variables the number of possible branches in the underlying branch and bound tree is also reduced as a result.

5.3.1.1 Modelling of Teaming constraints

Teaming constraints, as discussed in section 2.2.7, involve two or more employees when performing an activity. An approach to tackle such constraints is to use time-dependent synchronisation constraints. A synchronisation constraint allows two activities to start at the same time, each of them with its own employee. In general, the synchronisation constraints restrict the commencement of two activities, but allow for different finish times. The two activities can then finish at different times. By starting at the same time they cannot use the same employee. Using this concept, Teaming constraints can be modelled by creating virtual duplicate activities. A virtual activity is an exact replica of the original activity, i.e. same time window restrictions, same duration, same skills requirements, same location, etc. The number of virtual activities to be created is the required number of employees in the Team minus one. After these virtual activities are introduced, synchronisation constraints are added between every pairing of the original activity with a virtual one. Such approach guarantees that different employees are scheduled at the same start time to perform the original activity (through the virtual ones). For example, if activity A_0 requires a Team of three employees, two virtual activities are created A_1 and A_2. Then two synchronisation constraints are incorporated, i.e. A_0's start time = A_1's start time and A_0's start time = A_2's start time. A third constraint could be included, A_1's start time=A_2's start time, although it is clearly redundant.

5.3.2 MIP Model description

In this section an adaptation of the model by Rasmussen et al. (2012) is presented. The model introduces a new set of binary variables and changes the scheduling integer variables to fractional ones. The additional variables indicate whether an activity is or is not performed regardless of the employee. Therefore the number of extra binary variables is given by the number of activities. More variables might be required when an activity needs more than one employee, since that creates virtual activities. Adding these variables allows the model to mark activities as unassigned whilst maintaining feasibility of the solution.

$$\min \qquad \omega_1 \sum_{k\in K}\sum_{i\in N^k}\sum_{j\in N^k} C_{ij}^k x_{ij}^k + \omega_2 \sum_{k\in K}\sum_{i\in C}\sum_{j\in N^k} \delta_i^k x_{ij}^k + \omega_3 \sum_{i\in C} \gamma_i y_i \tag{5.12}$$

$$s.t. \qquad \sum_{k\in K}\sum_{j\in N^k} x_{ij}^k + y_i = 1 \qquad \forall\, i \in C, \tag{5.13}$$

$$\sum_{j\in N^k} x_{ij}^k \le \rho_i^k \qquad \forall\, k \in K, \quad \forall\, i \in C, \tag{5.14}$$

$$\sum_{j\in N^k} x_{0^k,j}^k = 1 \qquad \forall\, k \in K, \tag{5.15}$$

$$\sum_{i\in N^k} x_{i,n^k}^k = 1 \qquad \forall\, k \in K, \tag{5.16}$$

$$\sum_{i\in N^k} x_{ih}^k \min \sum_{j\in N^k} x_{hj}^k = 0 \qquad \forall\, k \in K, \quad \forall\, h \in C, \tag{5.17}$$

$$\alpha_i \sum_{j\in N^k} x_{ij}^k \le t_i^k \le \beta_i \sum_{j\in N^k} x_{ij}^k \qquad \forall\, k \in K, \quad \forall\, i \in C \cup \{0^k\}, \tag{5.18}$$

$$\alpha_{n^k} \le t_{n^k}^k \le \beta_{n^k} \qquad \forall\, k \in K, \tag{5.19}$$

$$t_i^k + s_{ij}^k x_{ij}^k \le t_j^k + \beta_i (1 \min x_{ij}^k) \qquad \forall\, k \in K, \quad \forall\, i,j \in N^k, \tag{5.20}$$

$$\alpha_i y_i + \sum_{k\in K} t_i^k + p_{ij} \le \sum_{k\in K} t_j^k + \beta_j y_j \qquad \forall\, (i,j) \in P, \tag{5.21}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall\, k \in K, \quad \forall\, i,j \in N^k, \tag{5.22}$$

$$t_i^k \in \mathbb{R}_+ \qquad \forall\, k \in K, \quad \forall\, i \in N^k, \tag{5.23}$$

$$y_i \in \{0,1\} \qquad \forall\, i \in C. \tag{5.24}$$

The constraint requiring all activities to be performed (constraint 5.2 in the IP model) is changed to include the binary variables y_i which indicate if activity i is assigned to an employee (constraint 5.13). A value of 1 indicates that the activity is left uncovered, whereas a value of 0 means the activity has been assigned. Continuous variables t_i^k, hold activity i's start time by employee k, these are made real positive values instead of integer ones as previously discussed (constraint 5.23).

The set C represents customer locations. Constant O^k holds the starting location of employee k and n^k corresponds to the ending location for k. This configuration supports different start and end locations for each employee. The employees' set is K. $N^k = C \cup \{0^k,\, n^k\}$ is the set of available locations for employee k. Notice how it includes his starting and ending location. Time window restrictions on activity i are given by α_i for earliest start time and β_i for latest start time. The value $\rho_i^k = 1$

indicates if employee k can perform activity i in terms of skill matching. Nevertheless, it can also be used to forbid an allocation between employee and activity. Travel time between two locations i and j is given by s_{ij}^k for employee k. If the data contains different travel time for every employee depending on his means of transportation, then each s_{ij}^k may be different. These variables already include the duration of activity i taking into account the experience employee k might have. If we assume that times are the same regardless of employee, we can use a single variable for all employees. If activities' duration are the same regardless of who performs them, then a single value can be used (s_{ij}) instead of one for each employee. Employee k starting time is given by α_{n^k} and finishing working time by β_{n^k}. All connected activities (synchronisation, overlap, minimum, maximum, min-max) constraints are given in set P. Every member of P is a pair of activities i and j subject to a type of time-dependent constraint given by a constant value p_{ij}. The values for p_{ij} are assigned according to Table 5.5. Finally, binary variables x_{ij}^k are set to 1 if employee k travels from location i to j and set to 0 otherwise (5.22).

Type	p_{ij}	p_{ji}
Synchronisation	0	0
Overlap	-(duration j)	(duration i)
Minimum difference	minimun difference value	not applicable
Maximum difference	not applicable	-(maximum difference value)
Min-Max difference	minimun difference value	-(maximum difference value)

Table 5.5: Values assigned to p_{ij} and p_{ji} obtained from (Rasmussen et al., 2012)

The objective function (5.12) is integrated by the cost C_{ij}^k, the preference δ_i^k that employees (k) have when performing activities (i) and the priority of the visit i given by γ_i. Cost C_{ij}^k could be defined depending on what is needed to be reduced, i.e. distance, time, money, etc. Priorities γ_i provide a way of differentiating important activities from those which might be left unassigned for another period. The weights $(\omega_1, \omega_2, \omega_3)$ can be adjusted to give more relevance to any of the three components of the objective function.

Constraint 5.13 means that visits are either performed or left unassigned. Activities can only be assigned to employees with the skills to perform them (constraint 5.14). Employees must start from their initial location and return to their final location (constraints 5.15 and 5.16). Constraint 5.17 ensures employees cannot stay at a customer location and forces them to leave until they reach their final location, i.e. flow conservation. Time windows of visits must be satisfied (constraints 5.18 and 5.20). All activities performed by employee k should start and end during his starting and ending working times (5.19). Every type of connected activities generates a constraint that is enforced by equation 5.21. Scheduling variables are restricted to be real positive since they capture the starting time of activities (5.23). Finally, if an activity i is not performed binary decision variables y_i is set to 1 and 0 otherwise.

5.3.2.1 Reduction in the number of variables

A mathematical model that uses fewer variables and still produces the same results is generally considered a better one. The same can be said about the constraints: a tighter representation that still enforces all the constraints is preferable. The number of experiments in section 5.2.3.2 in which the solver ran out of memory (see Table 5.1 for details) represented 25% of the used instances. In the MILP model the number of variables is increased due to the addition of the binary variables capturing if an activity is or is not performed. As a result, a mechanism that could reduce the number variables and tighten the model was sought. The solution came in the representation of activities that required two or more employees.

The modelling strategy for Teams has been explained earlier (see 5.3.1.1), the addition of virtual activities results in additional segments in the network. Such segments need a binary variable per employee to indicate whether the segment will be utilised (x_{ij}^k). The segments connect all activities (nodes) to each other, it is the constraints that prevent some segments from being used. The segments connecting an original activity to its virtual activities and vice versa can be omitted. The rationale involving such action considers that if an employee is performing the original activity then it is clear that he cannot be moved to a virtual activity since it represents the same. As a result, even though employees are able to reach and leave both the original and virtual activity, it cannot happen that they leave the original to go to a virtual one or vice versa. The prohibition is achieved by never creating extra binary variables (x_{ij}^k) between neither the original and the virtual ones, nor the virtual ones themselves. Such approach is represented in Figure 5.2.

5.3.3 Experiments

5.3.3.1 Setting parameters and weights in the MILP

In this set of experiments the MIP model presented in section 5.3.2 is used to solve the 375 instances in all five data sets. The solver time limit is set to two hours, based in the value discussed in Figure 5.1.

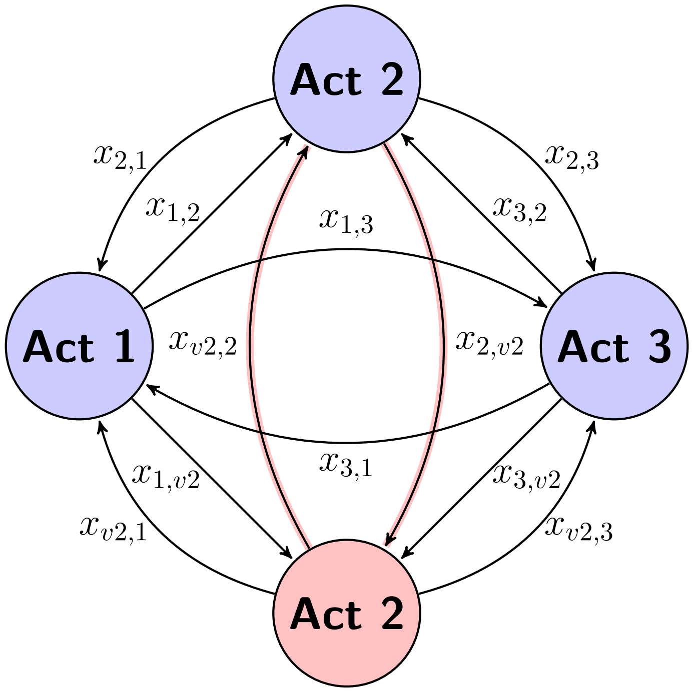

Figure 5.2: Activity 2 in red represents a virtual node. Segments in the network $x_{2,v2}$ and $x_{v2,2}$ are removed from the model, since once an employee enters either the original or virtual activity 2, it cannot move to the other. Therefore, it is possible to reduce the number of binary variables used. Prohibited segments are highlighted in red

The cost value C_{ij}^k in the objective function is calculated as travel time plus distance from i to j to make dependent on the location of the activity and its duration. C_{ij}^k differs from the previous cost c_{ik} in the IP Model. In the IP model c_{ik} depends on the activity i and the employee k. Whereas in the MIP model the cost C_{ij}^k varies depending the starting point i and the ending point j and the employee k. The latter representation is more flexible since it can represent diverse means of transportation for the segment, or some relationship between the origin and destination.

The weights $(\omega_1, \omega_2, \omega_3)$ are calculated as in Rasmussen et al. (2012). Such weights favour the allocation of all activities when possible by setting $\omega_3 > \omega_2 > \omega_1$. Assigning all activities is no longer a hard constraint but a soft one, it therefore needs to be included in the objective function. Nevertheless, if possible, the model should seek to prefer all activities to be allocated. As a result, the penalty of unassigned activities should be the biggest one (see 5.27). The second in importance is employees' preferences component penalty (see 5.26). Finally, the third factor is related to the cost with a weight of 1 (see 5.25). The values are given by the expression:

$$\omega_1 = 1 \tag{5.25}$$

$$\omega_2 = \sum_{k \in K} \sum_{i \in N^k} \sum_{j \in N^k} C_{ij}^k \tag{5.26}$$

$$\omega_3 = \omega_2 |C| max_{k \in K, i \in C} \delta_i^k \tag{5.27}$$

5.3.4 Results

This section presents an analysis of the results achieved by the mathematical solver. The analysis classifies the results of the solver in five categories and matches such results with the following characteristics of the data sets: number of employees, number of activities, ratio activities/employees, time window size, ratio planning horizon/mean activity duration, activities requiring teams and time-dependent constraints. In addition, graphs summarising the percentage gap are also discussed. The reader is referred to Tables C.1 and C.2 in Appendix C which show individual instance results provided by the mathematical solver. Both tables include best objective value, best lower bound obtained, the gap reported by Gurobi, the computation time and the category assigned for each of the instances.

The results of the mathematical solver can be classified in five categories. The first category, "Unloadable", is assigned to instances that due to the size of the instance, i.e. number of variables and constraints, the solver was unable to load the model as it runs out of memory during the loading process. The second category, "OutOfMemory", relates to instances in which the solver starts the optimisation process but runs out of memory before the two hours of computation time. The solver provides neither a lower bound nor a feasible solution. The third category, "No solution", includes those instances where the solver completes the time limit, a lower bound was found but no feasible solution is reported. The fourth category "Optimal", is where the solver reports optimality. Finally, the last category "Non-Optimal" is for instances for which a feasible solution (non-optimal), a lower bound and a gap are provided by the solver. The 375 instances are distributed among the five categories in the following manner: 6 are "Unloadable", 13 in "OutOfMemory", 18 have "No Solution", 34 included in "Optimal" and 304 in "Non-Optimal".

The Unloadable category contains instances with the largest number of activities and employees as all have 250 activities and 171 employees. Instances in the OutOfMemory category present 150 or more activities but less than 250. They also have more than 41 employees. They are part of Mov and Sec data sets. The instances in the No

Solution category have between 90 and and 201 activities and more than 20 but less than 50 employees. All instances in this group belong to Sec. The Optimal category includes instances with up to 100 activities and a maximum number of employees of 38. Optimal instances are the majority in Sol data set but there are five of Mov and one of HHC. The instances for which feasible non-optimal solutions are found (Non-Optimal category) vary from 25 activities up to 201 activities with a average of 82, and number of employees ranging from five employees to 50 with a mean of 19.

5.3.5 Analysis of results

Figure 5.3 provides five box plot graphs, each referring to one of the five categories defined for the MILP results. The graphs consider the number of employees for each category. A clear tendency can be seen as the median number of employee for each category increases with respect to the previous categories. The categories are ordered depending the type of result. In general, optimal solutions are achieved for instances where the number of employees is small. Instances that belong to the Out of Memory and Unloadable categories have the largest number of employees.

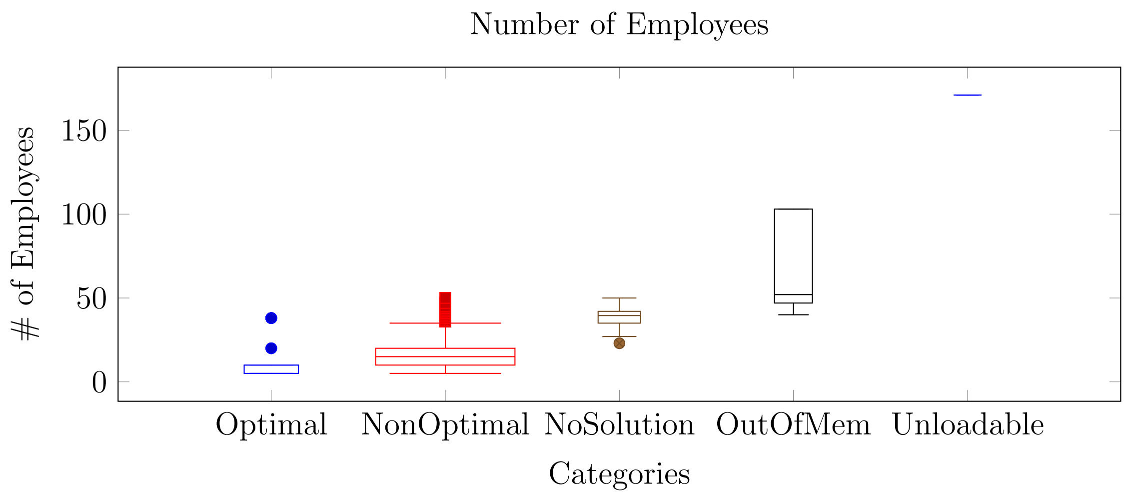

Figure 5.3: Box plot showing lower quartile, median and upper quartile on the number of employees for each of the five categories defined (optimal, non-optimal, no solution, out of memory and unloadable)

Figure 5.4 illustrates how the number of activities in the instances are distributed among the five different categories of results. Similar to number of employees, the figure shows a tendency of increase in the median value of activities relating to the results of the solver. The majority of optimal instances have fewer than 50 activities and those for which the solver could find a feasible non-optimal solution are concentrated between 50 and 100 activities. The previous two categories present outliers. It is worth noting how close the instances of the No Solution group are to the Out of

Memory one; the median is almost the same. It can be said that since instances in the No Solution category have not found any solution after two hours, it is more likely that if more time is given to the solver they might result in running out of memory. As with the number of employees, it seems a large number of activities also makes instances unloadable for the solver.

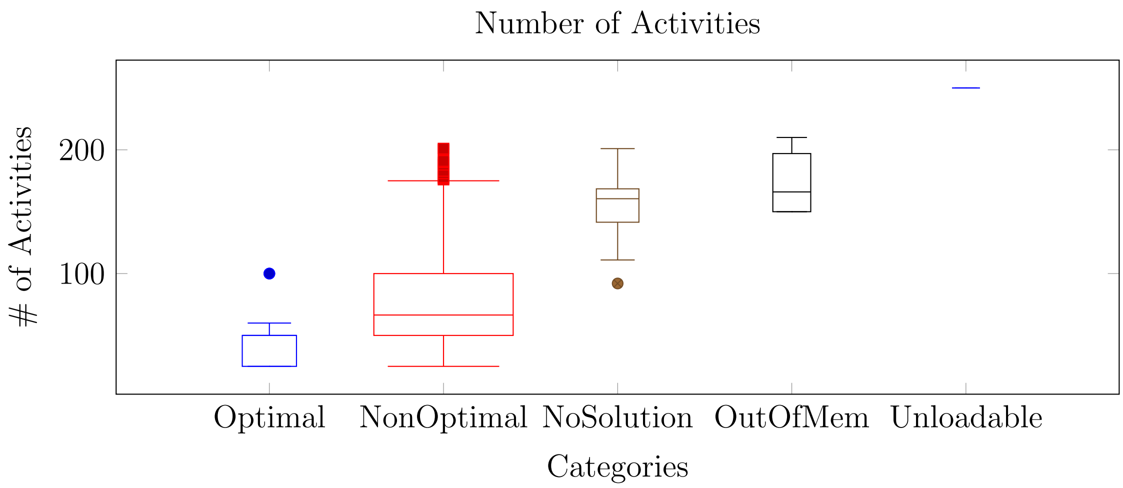

Figure 5.4: Box plot showing lower quartile, median and upper quartile on the number of activities for each of the five categories defined (optimal, non-optimal, no solution, out of memory and unloadable)

Figure 5.5 shows the distribution of the ratio between the number of activities and the employees available to perform them for each category. A ratio of five activities per employee is the median value for optimal instances and from there the tendency seems to be the lower the ratio the more difficult for the solver. This could be explained as a lower ratio indicates more employees for the same number of activities, it requires the solver to evaluate more possibilities. But given the categories Out of Memory and Unloadable medians, it can be said that knowing the ratio by itself does not help if at least another value, e.g. number of employees or activities, is not known. For example in a small instance with 20 activities and two employees the ratio is the same as one instance with 200 activities and twenty employees, i.e. ratio is 10. However, the one with 200 activities might not be able to be loaded by the solver.

Figure 5.6 shows the distribution of the mean time window size across the five categories. Although when observing the figure the longer the mean time window size the more difficult the instances, there are a few more outliers compared to the previous figures. A larger mean time window size means there are more possibilities to assign a different starting time, which can be seen as more options to consider, therefore more difficult for the solver. But, small mean time windows size can also be difficult because it restricts the search. The box plot uses the mean time window size, but since the mean does not provide a sense of dispersion, it is left for further investigation the

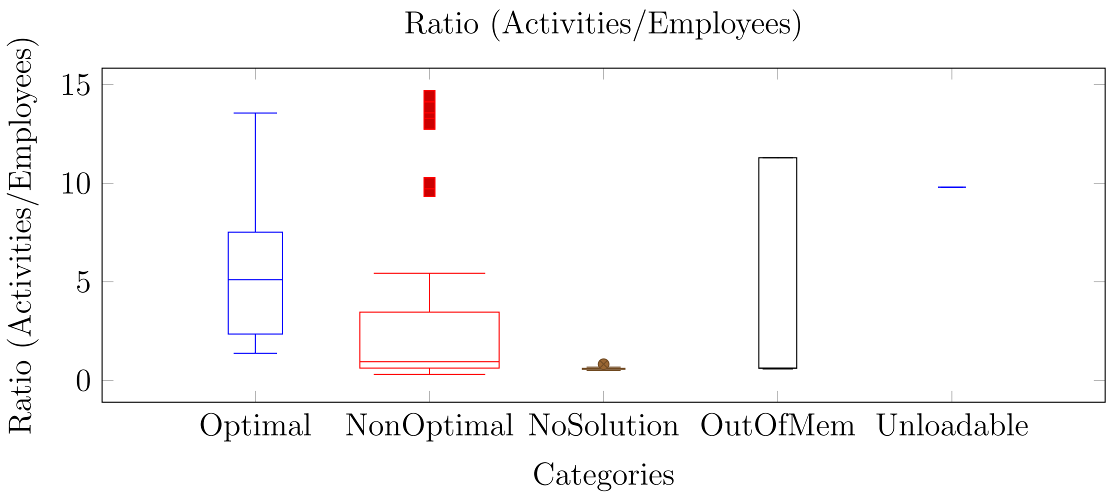

Figure 5.5: Box plot showing lower quartile, median and upper quartile on the ratio (activities/employees) for each of the five categories defined (optimal, non-optimal, no solution, out of memory and unloadable)

case of whether the same time window for all activities could be easier to solve than some activities presenting large, medium and small time window sizes but both with the same mean. A similar investigation on the distribution of different time window sizes for the VRPTW found evidence for such a case (Castro-Gutierrez et al., 2011).

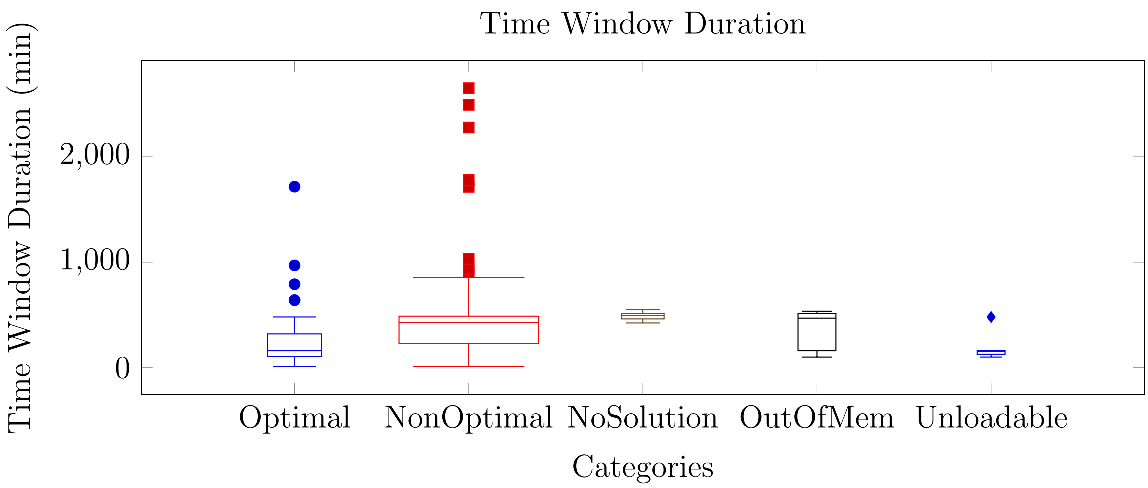

Figure 5.6: Box plot showing lower quartile, median and upper quartile on the average duration of time window for each of the five categories defined (optimal, non-optimal, no solution, out of memory and unloadable)

Figure 5.7 presents another ratio distribution. The ratio between the planning horizon of an instance and the mean duration of activities. A smaller ratio as observed makes the instance more difficult for the solver, but more information is required in order to contextualise this ratio. The ratio itself does not require knowing number of activities or employees in order to be calculated. If these two features remain unchanged a reducing ratio means more work due to three possibilities. The first possibility is the increase in the mean duration of activities. The second possibility is the shortening of the planning horizon. A third possibility considers a combination of the previous two.

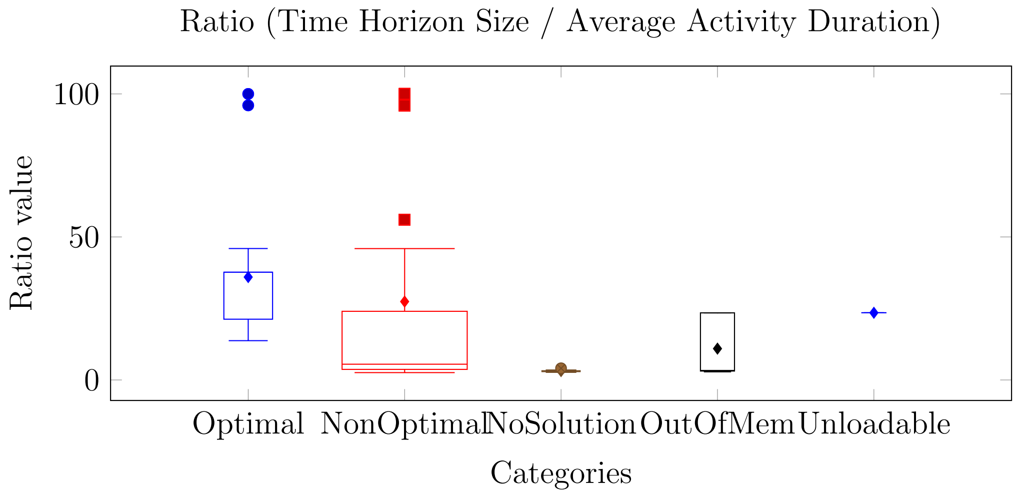

Figure 5.7: Box plot showing lower quartile, median and upper quartile on the ratio (planning horizon duration/mean activities duration) for each of the five categories defined (optimal, non-optimal, no solution, out of memory and unloadable)

Figures 5.8 and 5.9 show the distribution of teams and time-dependent activities constraints across the five categories. Both figures are similar in structure since teams are implemented as time-dependent constraints between activities and their virtual counterparts. Overall the observation is that instances with more time-dependent constraints tend to be harder for the solver.

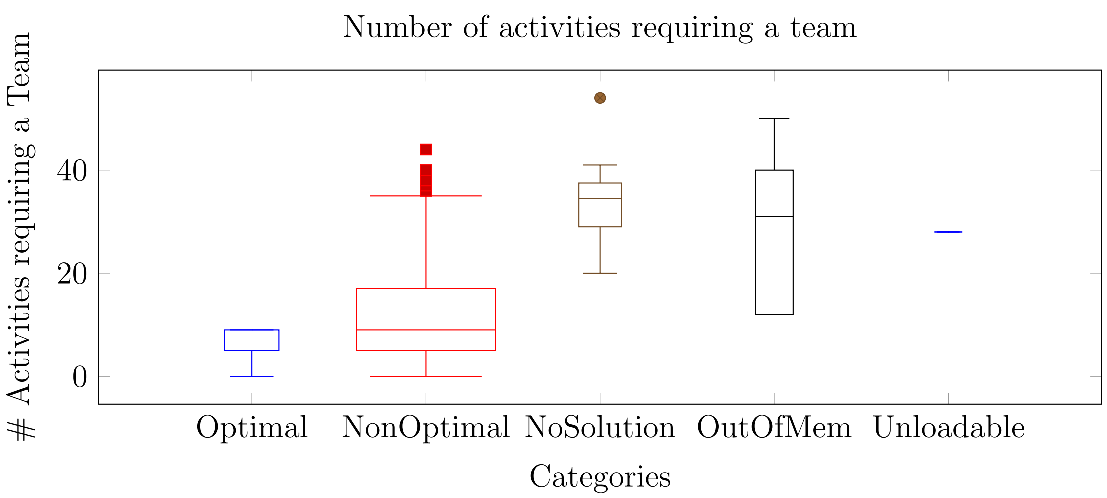

Figure 5.8: Box plot showing lower quartile, median and upper quartile on the number of teams for each of the five categories defined (optimal, non-optimal, no solution, out of memory and unloadable)

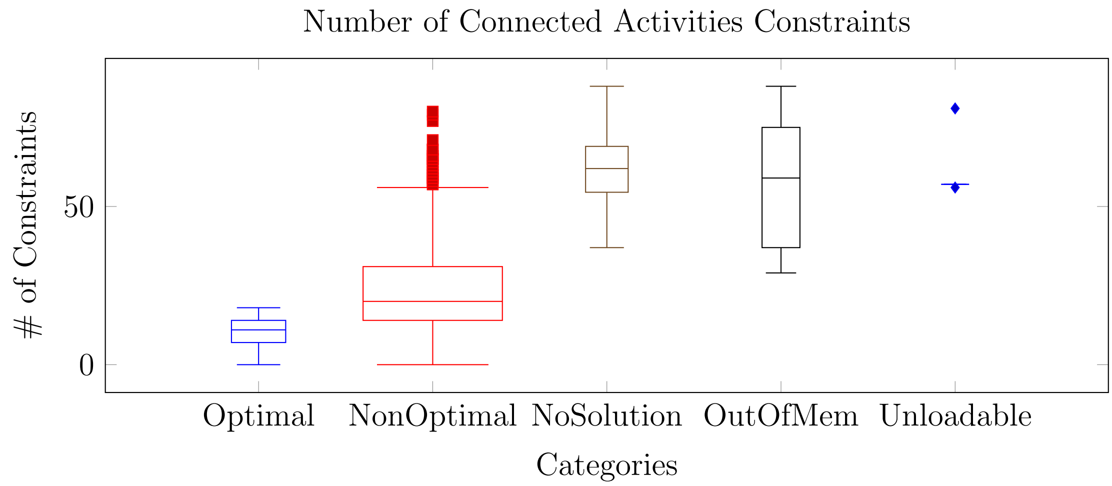

Figure 5.9: Box plot showing lower quartile, median and upper quartile on the number of time-dependent activities constraints for each of the five categories defined (optimal, non-optimal, no solution, out of memory and unloadable)

5.3.6 Gap analysis

Out of the 375 instances, the solver found feasible solutions for 338 of them (Optimal + NonOptimal categories). It is observed that the larger instances are part of the Unloadable category. These are five of the Mov data set (250 activities) and the single Tech instance. The instances in OutOfMemory are those with 150 or more activities which are not part of the Unloadable category, mainly belonging to Mov (150 activities) and Sec data sets. However, this is not a definitive indication of the difficulty of the instance with this model and solver because some scenarios with more than 150 activities were solved, although in those cases the number of employees was less than 100. The largest number of activities in an instance that the solver found a solution for was 200 activities and 50 employees.

We identify groups of instances based on the number of employees, as shown in Figure 5.10, where the number of instances in each group is given. For example, HHC_9 refers to the eight instances from HHC data set that have nine employees. Because the Sec instances have greater variety in the number of employees (see Figure 4.1) they are grouped in ten clusters (from Sec_10 to Sec_55, adding five employees for every new cluster). Each instance is assigned to the closest upper cluster, e.g. instances in Sec with six or seven employees are assigned to cluster Sec_10 and instances with eleven employees are assigned to cluster Sec_15.

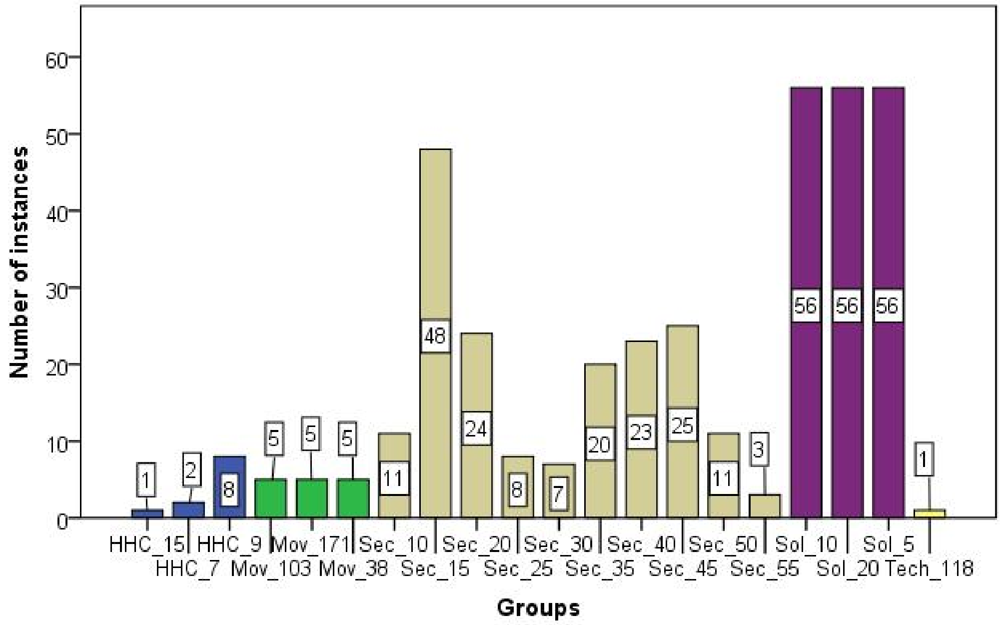

Figure 5.10: Groups of instances based on number of employees.

5.3.6.1 Analysis of Feasible Results

During the optimisation process, the solver provides information about the gap between the lower bound and the current feasible solution (if it exists). Such gap for the 338 solved instances is shown in Figure 5.11, where it can be observed that there is a widespread range of values across the instances. The figure uses an identifier of each instance in the horizontal axis, these identifiers are allocated according to the data set they belonged to: Sol [1, 168]; Sec [169, 348]; Mov [349, 363]; HHC [364, 374], Tech [375], some indexes are not used since they belong to instances in categories (Unloadable, Out of Memory and No solution). It is noticeable that optimality was achieved for some of the Sol and Mov instances (gap value of zero). However, for Sec the solver reported a gap of more than 50% in most of the cases (see Fig. 5.12).

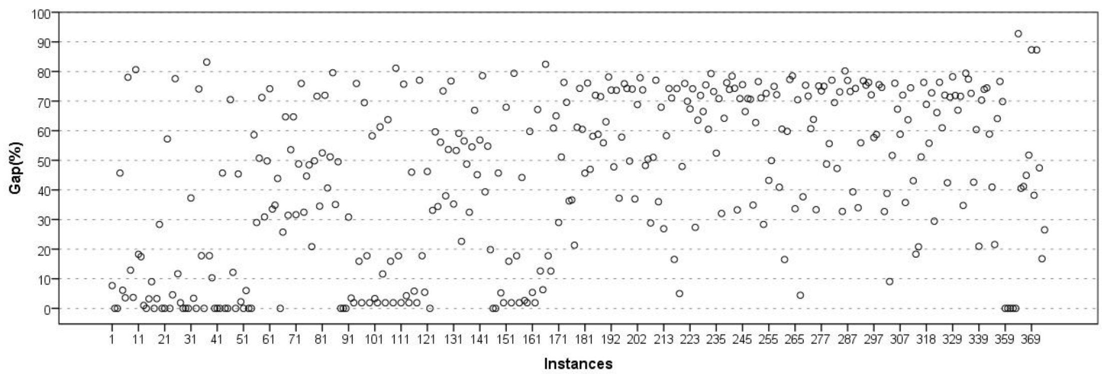

Figure 5.11: Percentage gap values for 356 instances where the solver found feasible solutions.

In order to achieve a better understanding of the difficulty of the WSRP, the gap results for the instances in each data set are split. Figure 5.12 shows the gap results for the instances by data set. It is clearer that the Sec instances are more difficult to solve given the larger number of high gap values reported. This can be confirmed in the corresponding box plot in Figure 5.13 which shows a median of 67%. However, low gap values were reported for the Sol instances with a median of just above 20%. For the mayority of HHC instances the gap value is between 40% and 70%. The results shown for the Mov data sets are for the five instances with 50 activities and 38 employees. For all of them an optimal solution was found.

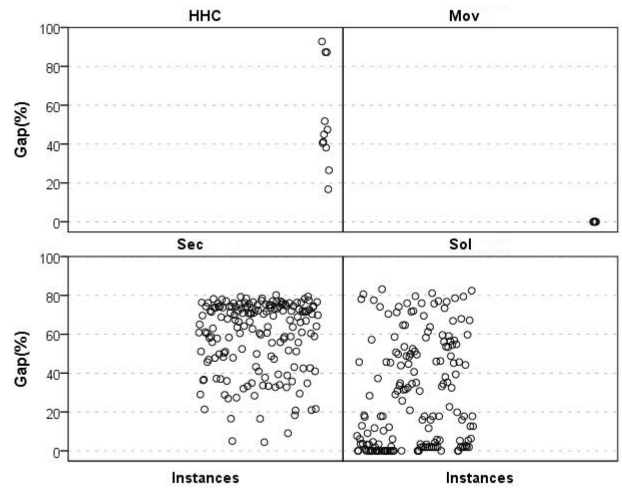

Figure 5.12: Distribution of gap percentage values reported for each instance in each data set.

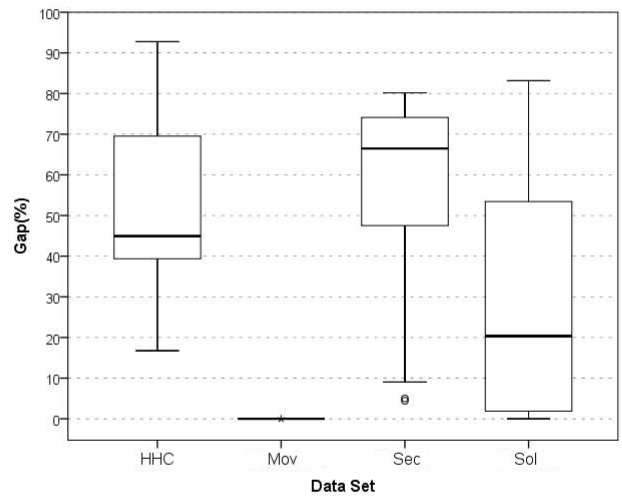

Figure 5.13: Aggregated gap percentage values reported for the instances in each data set.

Figure 5.14 plots the gap values reported for all instances within different clusters with respect to the number of employees. Looking at the three HHC groups it is clear that the smaller the size of the workforce, the better the gap value achieved. A similar observation can be made for the Sec instances: the achieved gap value worsens as the number of employees increases. Note that this tendency is not clear for the Sol instances. Looking at the box plot of the Sol group in Figure 5.15 we can see that 50% of the instances with ten employees (Sol_10) report a gap value below 5%. Also, 50% of the instances with five employees (Sol_5) report a gap of 15%. That is, more instances in Sol_10 report a better gap than instances in Sol_5.

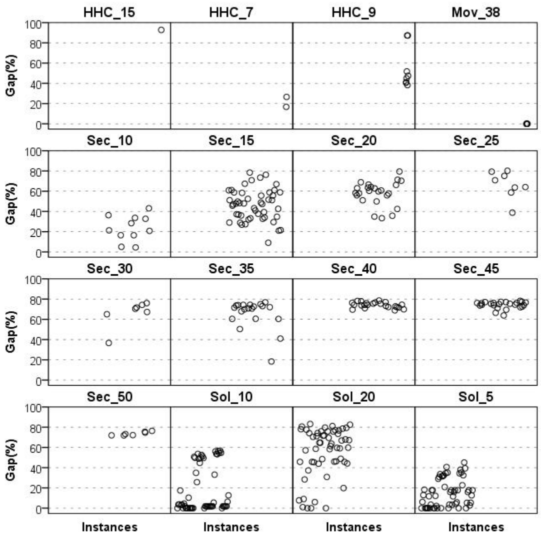

Figure 5.14: Distribution of gap percentage values reported for each instance grouped by data set and number of employees.

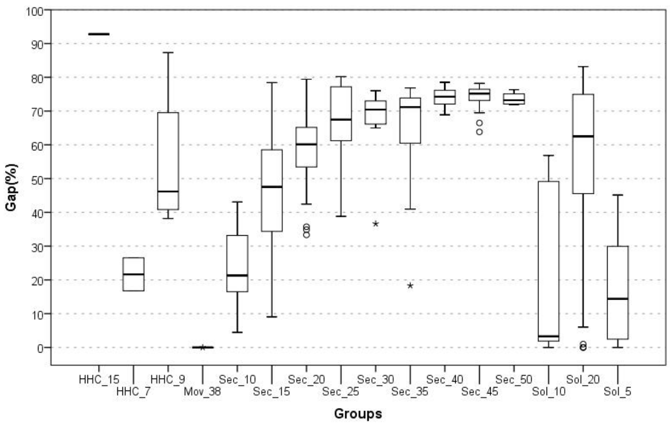

Figure 5.15: Aggregate gap percentage values reported for each group of instances according to the number of employees.

5.4 Conclusions

The change from a IP to a MIP model helped to obtain a better understanding of WSRPs. The MIP model allows activities to be unassigned which can provide feasible results for understaffed instances. Such approach is needed, for example, when reassigning low priority activities in order to cover an emergency one. Even though activities can be unassigned, it does not mean that this is desirable. As a result, it is necessary to apply the biggest weight for the unassigned component in the objective function. By doing so, the mathematical solver is forced to try as much as possible to assign all activities as unassigned activities come at a huge penalty. The new objective function includes employees' preferences.

Regarding the results, it can be concluded that integer feasible solutions are found for instances with up to 200 activities. The number of activities and employees influences the difficulty encountered by a mathematical solver. Either of the two values is required when contextualising both ratios (activities/employees and planning horizon/mean activity duration) as measures of difficulty for the solver. The more teaming and time-dependent activities constraints there are in an instance makes it harder for the solver to tackle. Gap percentages, between the best solution reported and the lower bound found by the solver, are smaller overall in the Sol data set than in the Sec one. The best objective values acquired, for almost all instances (338 only), establish a benchmark for future solution methods. Future methods should attempt to improve their quality of results via: finding better objective values for the minimisation problem; and/or, reducing the amount of time required to obtained the same quality level of results reported in this chapter's benchmark.

Chapter 6

A Greedy Heuristic for WSRP

6.1 Introduction

In the previous chapter, exact methods were presented using two mathematical models. In this chapter, a new solution approach to tackling WSRP is discussed: a Greedy Heuristic (GH). This new solution approach is based on heuristic methods which do not guarantee finding the optimal solution for a problem, but are widely used when tackling hard combinatorial optimisation problems. One advantage is that they are tailored for each domain problem. Heuristic methods can use more information to strategically drive the search. The disadvantages often faced by heuristic methods are that they might get trapped in local optimum when searching for better solutions, and they cannot guarantee finding optimal solutions. Therefore, it is important to have benchmark results from the mathematical solver, because such results allow comparison between the exact methods and the heuristic ones (Rees, 1996; Burke and Kendall, 2005).

Greedy heuristics are procedures based on consecutive decisions that where possible lead to progressively better results until no further improvement can be achieved. The term Greedy Heuristics can be used interchangeably with Constructive Heuristics, or Successive Augmentation Algorithms (Talbi, 2009, pg. 26). Greedy heuristics rely on some information about the domain of the problem and often obtain good feasible results in a short time. However, greedy heuristics do not guarantee finding optimal solutions. In some cases, the results obtained by them are far from the optimum one. In fact, even if an optimal solution is found, greedy heuristics will not know is optimal. The greedy heuristic presented in this chapter is deterministic, i.e. it provides the same result as long as the instance and parameters are the same (Burke and Kendall, 2005, pg. 10). The last version of the heuristic introduces a random choice which is controlled.

This chapter describes a deterministic greedy heuristic (GH) which developed from a bin-packing problem analogy. Five versions of GH are discussed. For every version of GH, experiments are presented using all 375 instances. The result of the experiments is analysed and improvements to the next version justified. GH was designed iteratively and every version builds on the improvements of the previous ones. Before starting the description of GH the data structure used for all versions is presented.

6.2 Solution Structure Representation

In this section the solution representation of a WSRP is discussed. A solution to the WSRP must provide the subset of activities that each employee performs along with the order in which the activities are executed. It should also indicate the number of activities that are left unassigned, if any.

The data structure proposed for the WSRP is represented in Figure 6.1. The main array contains nodes of employees. The number of employees is defined in the problem instances. The configuration resembles the matrix configuration proposed by Mankowska et al. (2014). In addition, there is one extra node in the main array, the node holding unassigned visits. Every employee-node contains a list of visits which the employee is assigned to perform. The employee's list of visits is kept in ascending order according to the earliest starting time of the activities. In the figure, green rectangles represent activities' durations, gray rectangles depict travelling time between locations, and idle time is shown in yellow rectangles. The last blue rectangle on each list of visits represents the travelling time back to the employee's final destination. Activities in the unassigned array node have no particular order. The employee's list of activities contains two types of nodes. The first type is scheduled activity nodes. Each scheduled activity node contains information regarding its starting time, ending time, duration, and time window restrictions. The second type is travelling nodes which hold the starting and ending travel time between two locations. Idle times are not stored in the data structure. Idle times are computed when required. For every scheduled activity node there is one travelling node which has an ending time equal to the starting time of the scheduled activity. The decision to store travelling time nodes between scheduled activity nodes is in order to reduce computation and to facilitate constraint evaluation.

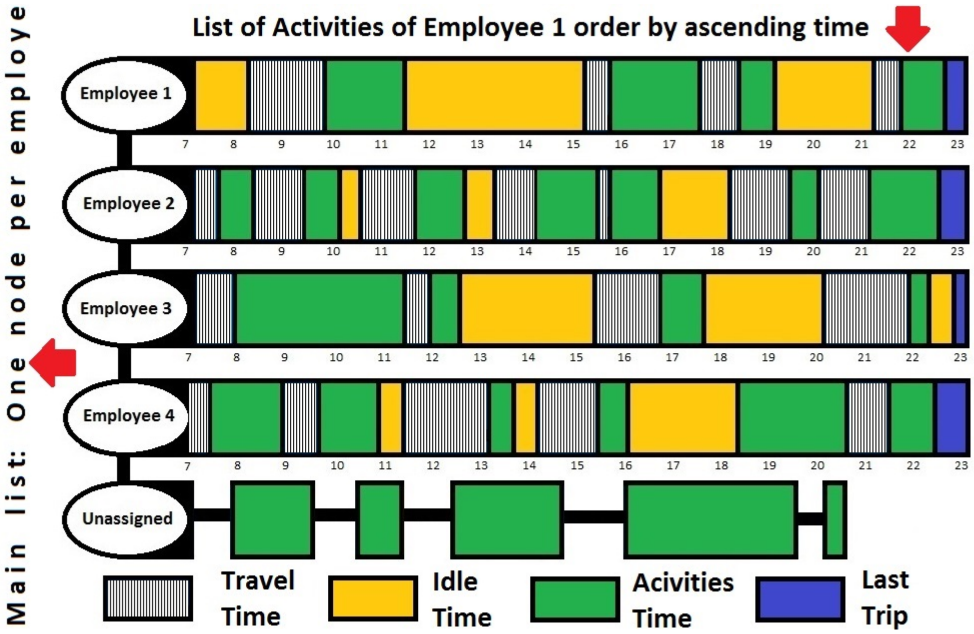

Figure 6.1: Solution structure for WSRP: an array of lists. The main array contains nodes for every employee in the workforce plus an additional node for the unassigned activities. Every employee has a list of activities. Lists are kept in order according to the activities' starting time except the one where unassigned activities are held i.e. it is a set.

6.3 Design of the Greedy Heuristic

The final version of GH included the improvements obtained iteratively from its inception. In this section the major differences between versions of GH are discussed. Every change performed was aimed at improving the quality of the final solution. The GH initial solution procedure was inspired by the bin-packing problem (Schwerin and G., 1997).

6.3.1 Bin-packing inspired approach

The first version of GH, henceforth referred to as GH1, is based on the bin-packing problem. In the bin-packing problem there are a number of objects with different dimensions. The objects need to be packed into a finite number of bins. Every bin has limited capacity, once one bin is full another bin needs to be used. The bin-packing problem objective is to minimise the number of bins used when packing all the objects.

GH1 considers the activities as one-dimensional objects and every employee as a bin. Employees' working time can be seen as bins' capacity. After assigning all activities across all employees, each bin holds the activities of one employee. An additional bin is required to keep the unassigned activities. Using the bin-packing analogy in WSRP relates to the concept of an employee's "full" schedule.

The analogy makes it simple to understand, although applying it to WSRP means there are several issues to consider. One issue is that not all activities could be assigned to all employees due to the skill-related restrictions, whereas in the classic bin-packing problem all bins can be used as long as they have space (Dawande et al., 2000). However, there are versions of bin-packing with assignment restrictions. Another issue with GH1 is that the representation of activities' start time is not taken into account. As long as the duration of all activities in a bin fits into the employees working time, i.e. the activities are "packed" into the employee, the solution is feasible, but such an approach cannot work for WSRP. As a result, the activities have to store their own starting time. The next issue is that activities within bins (employees) have to remain sorted to guarantee that the ascending order of start and end times is maintained. Activities can be sorted before starting the assignment process using their minimum starting time. The final issue is enforcing the time windows and time-dependent constraints. Therefore, GH1 does not address the time-dependent constraints. Time windows are enforced when selecting the employee that the activity should be assigned.

Algorithm 1 Greedy Heuristic 1 (GH1)

```
procedure Solve
    visitList ← copy(visits)
    sol ← NewSolutionStructure()
    Sort(visitList, listCriterion)
    while ¬empty(visitList) do
        Sort(sol, solCriterion)
        v ← remove(visitList,0)
        can ← AllocPossible(v,sol)
        Sort(can)
        if ¬empty(can) then
            c ←remove(can,0)
            Include(c,sol)
            i ← v.required
            for  i > 1 do
                if ¬empty(can) then
                    c ← remove(can,0)
                    Include(c,sol)
                else
                    Unallocate(v,sol)
        else
            Unallocate(v,sol)
```

```
function AllocPossibleAny(v,sol)
    for n ← sol.nodes do
        e ← n.emp
        if ¬ Perform(e,v) then
            next
        if ¬empty(n.sch) then
            w ← LastAvWindow(n.sch)
            ca ← Enough(w,v.win,e)
            if ¬nil(ca) then
                add(can,ca)
        else
            for w ← 1, n.sch do
                ca ← Enough(w,v.win,e)
                if ¬nil(ca) then
                    add(can,ca)
            w ← LastAvWindow(n.sch)
            ca ← Enough(w,v.win,e)
            if ¬nil(ca) then
                add(can,ca)
```

Algorithm 1 provides the pseudo-code of the GH1. This algorithm changes in later versions (GH2 to GH5) but there are components that remain the same throughout.

A description of the non-changing features is provided. In line (2) a copy of the visits is created and stored in the variable *visitList*. An empty solution structure is created by using the NewSolutionStructure procedure (line 3). An empty solution only has employee nodes. The variable *visitList* is sorted according to *listCriterion*, a parameter explained later in section 6.3.1.2 (line 4). The sorting determines the order in which every visit is processed by the algorithm. The while loop (line 5) continues until *visitList* is empty, i.e. all visits have been either assigned or left unallocated. The main array order determines how the algorithm attempts to assign an activity to an employee. After each iteration, the solution structure *sol* is sorted according to *solCriterion*. The values available for the sorting criterion *solCriterion* are explained in section 6.3.1.2. The first visit is removed from *visitList* and kept in v (line 7). The procedure AllocPossible (Algorithm 1) finds possible allocations for visit v in the solution structure *sol*. The procedure ensures that candidate allocations are for employees with the right skills to perform the activities, and that the activities can be fitted in the assigned employee schedule whilst enforcing time window constraints. AllocPossible verifies if v can fit between the last visit in an employee schedule and the end of his working time. The function returns a list of candidate allocations after searching *sol* (line 8).

Algorithm 2 Greedy Heuristic

```
function AllocPossible(v,sol)
    for n ← sol.nodes do
        em ← n.emp
        if Perform(em,v) then
            w ← LastAvWindow(n)
            cw ← Clash(v.win,w)
        else
            return false
        if cw.dur > 0 then
            r(v) ← ttAntNew + v.dur + dtt
        else
            return false
        if r(v) ≤ w.dur then
            s ← cw.st − ttAntNew
            e ← w.et − ttAntPos − v.dur − dtt
            cn1 ← s > w.st
            cn2 ← cw.st ≤ e
            if AND(cn1,cn2) then
                z ← w.st + ttAntNew
                if v.est > z then
                    st ← v.est
                    it ← st − z
                else
                    st ← w.st + ttAntNew
                    it ← 0
                z ← w.et − ttAntPos
                if v.lst < z then
                    ft ← v.lst − st
                else
                    ft ← z − st
                ca ← NewCa(st, it, ft, em)
                add(can,ca)
                return true
            else
                return false
        else
            return false
```

A candidate allocation is a data structure that holds information on where to insert an activity within an employee's list of visits. A candidate allocation contains: a reference to the employee; a proposed start time for activity v; flexible time by which the start time can be delayed and still be a valid allocation; and the idle time which represents the time wasted by the allocation if the proposed start time is used.

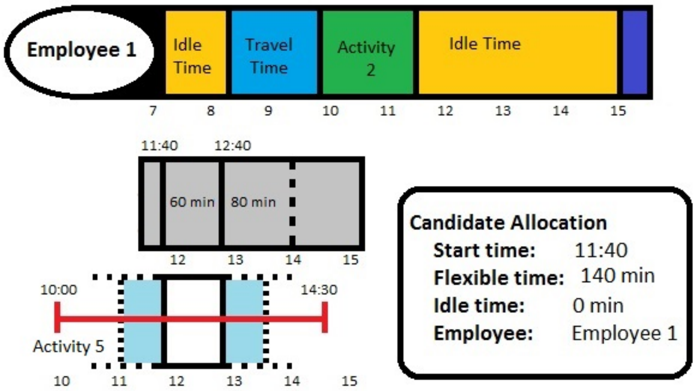

Figure 6.2: A candidate allocation contains: the proposed start time for an activity; flexible time by which the activity can be delayed for; the idle time before the proposed start time; and a reference to the employee for which the allocation can be applied.

Figure 6.2 illustrates a candidate allocation. Employee 1 has idle time from 11:30 until 15:00 as shown in the figure (second yellow rectangle). Activity 5, which duration is one hour, can fit within the available idle time. The activity's time window is set from 10:00 to 14:30 (represented in the red line). The idle time starts at the end of activity 2 (11:30), which is already in the employee's schedule. The travel time between activity 2 and activity 5 locations is 10 minutes. As a result, the start time of activity 5 can be set to 11:40. The flexible time considers both the time window of the activity and the remaining idle time. The latest start time of activity 5 according to its time window is 14:30. If such time is used, the activity would end at 15:30 which is beyond the end of idle time (15:00). Therefore, using the end of the idle time as a reference (15:00), activity 5's duration is subtracted from the end of the idle time. The result leaves 14:00 as the maximum start time of activity 5 that fits the idle time and adheres to the activity 5's time window. The difference between the maximum start time found and the start time established previously is 140 minutes, which becomes the value of the flexible time. The idle time component of the candidate allocation is set to zero since there is no wastage of time before the proposed start time of the activity.

After obtaining a collection of candidate allocations, the algorithm sorts the collection using the candidates' start time in ascending order (Algorithm 1, line 9). Other sorting criterial could be considered apart from the start time, e.g. idle time or flexible time. The next statement in the algorithm verifies that there are available candidates (line 10). If there are no candidates then v is left in the unassigned node by calling the function UNALLOCATE. If there are candidates, the first candidate is removed from the collection of candidates and kept in c. The candidate information is used to include v in *sol* by calling the function INCLUDE (lines 11 and 12). If v only requires one employee then the procedure moves to the next iteration. If on the contrary, more employees are required, a subsequent loop (line 14) iterates removing the next candidate available from *candidates* and including it in *sol* until all employees required are assigned (line 14 to 19). If at any time the candidate list becomes empty whilst still requiring employees then all previous allocations of v are removed and v is left in the unassigned node (line 19). All the candidates assigned within the inner loop need to have the same start time as the first assigned candidate, otherwise the team members will not be synchronised in order to start the activity.

6.3.1.1 Parameter *solCriterion* values

The *solCriterion* (Algorithm 1 line 6) parameter establishes the sorting criterion of the main array of employees. The possible values for the parameter are: one based on the remaining available time in an employee schedule and another based on the size of the list of activities within each node of the main array.

Remaining time sorts the solution list in descending order based on the time left available for every employee. The time available is calculated from the last visit until the end of the employees' shift. It aims to reduce the number of employees since it avoids using a new employee unless the available time of the previous ones are full or no other allocation is possible. If the sorting is ascending, activities are balanced across all possible employees with the right skills.

Solution size orders the nodes in the solution's main array in ascending order based on the number of visits that employees have assigned. When two nodes have the same number of visits, the second criterion is the remaining time left from the termination of the last visit until travelling back to the employees' final destination.

6.3.1.2 Parameter *listCriterion* values

The second parameter *listCriterion* (Algorithm 1 line 4) that can be changed within GH1 is the criterion for the initial sorting of the activities, i.e. the order in which they are processed by the heuristics. The possible values are described as follows:

Duration sorts visits in descending order based on the duration of the activity ($v.dur$).

Latest finish time sorts visits in ascending order based on the latest time the visit can finish according to its time window. The sorting parameter is obtained when adding *v.let*.

Latest start time sorts visits in ascending order based on the latest time the visit can start according to its time window. The sorting parameter is *v.lst*.

Earliest finish time sorts visits in ascending order based on the earliest time the visit can finish according to its time window. The sorting parameter is *v.eet*.

Earliest start time sorts visits in ascending order based on the earliest time the visit can start according to its time window. The sorting parameter is *v.est*.

Number of employees sorts visits in decreasing order based on the number of employees required. The parameter for sorting is *v.req*.

Density sorts visits in decreasing order based on the *density factor*. The density factor is obtained by adding the number of employees required plus the number of connected activities constraints that the visit is involved in. The aim is to process those activities which are complex first and leave the simple ones at the end. The number of employees in the team is modelled using synchronisation constraints.

6.3.2 Broadening the search for idle time: GH2

GH2 improves GH1 by broadening the scope of AllocPossible. The AllocPossible procedure in GH1 only searches available idle time between the last assigned activity and the end of the employee's working time. However, there might be other idle times within the employee schedule. Such "hidden" idle times appear as a result of enforcing time window in activities that cannot start earlier. Figure 6.3 shows a solution structure with three employees. If a new activity v is next for assignment the AllocPossible only evaluates idle times 3, 6, and 8. The improved version, a procedure called AllocPossibleAny (1) considers all idle times (1 - 8), thus increasing the options for assigning v. An updated version (GH2's pseudo-code) of the heuristic refer to appendix D.1.

Thus far, GH1 and GH2 have not addressed time-dependent activity constraints, although in many instances their presence makes solutions provided by GH1 and GH2 infeasible. The following section incorporates functions to deal with time-dependant activities.

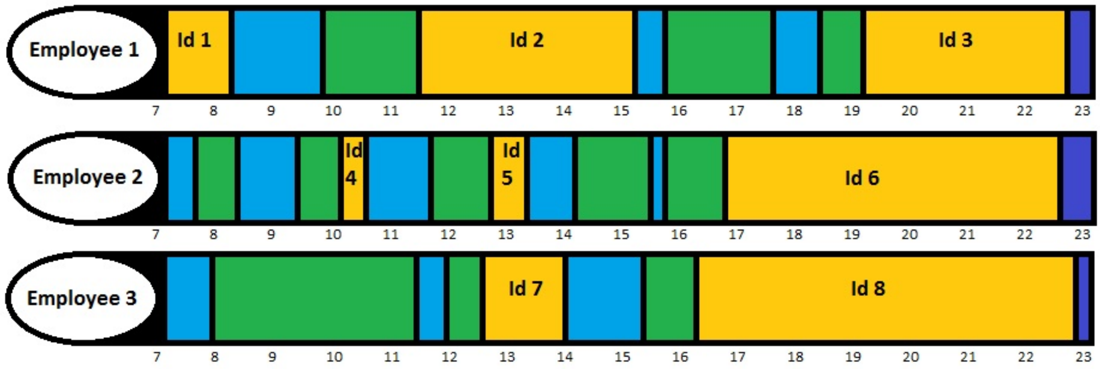

Figure 6.3: shows a set of idle time for a workforce consisting of three employees. Using AllocPossible considers only idle times 3, 6 and 8. The search can be broadened to include consideration of idle times 1, 2, 4, 5 and 7. This extension is perfromed by the AllocPossibleAny procedure.

6.3.3 Addressing time-dependent constraints: GH3

The improvements in this section tackle the handling of the time-dependent constraints. In order to apply such improvements it is necessary to differentiate when an activity is *independent* or *simple*, i.e. its start time is not restricted by any other constraint apart from its own time window, or *dependent* and *complex*. An activity becomes dependent if it requires knowing the start time of another activity in order to establish its own start time. There is the case of time-dependent activities. GH3 is shown in Algorithm 3. Lines 1 to 7 are similar to the Algorithm 1. The call to the Process method (line 8) delegates the assignment of visit v into the solution structure *sol*. The method Process returns a list of related visits (lrv), i.e. related by a time-dependent constraint. For every related visit of v (see while loop in line 10), the algorithm delegates the assignment of the related visit $v2$ to the ProcessDep method which updates lrv to remove the currently assigned and incorporate new related visits of $v2$ (line 12). Finally, $v2$ needs to be removed from the *visitList* as it has already been assigned.

Algorithm 3 Greedy Heuristic 3 (GH3)

1: **procedure** Solve
2:     $visitList \leftarrow$ copy(V)
3:     $sol \leftarrow$ NewSolutionStructure()
4:     Sort($visitList$,$listCriterion$)
5:     **while** ¬empty($visitList$) **do**
6:         Sort(sol,$solCriterion$)
7:         $v \leftarrow$ remove($visitList$,0)
8:         $l \leftarrow$ Process(v)
9:         **if** $l.size > 0$ **then**
10:             **while** ¬empty(l) **do**
11:                 $v2 \leftarrow$ Get(l,0)
12:                 $l \leftarrow$ ProcessDep($v, v2, l$)
13:                 remove($visitList$,$v2$)

Algorithm 4 shows both Process and ProcessDep. The Process method finds a list of candidate allocations by calling AllocPossibleAny (line 2). It sorts the candidates (line 3). If there are available candidate allocations by always removing the first one and considers it to include activity v in the solution structure (lines 4 to 6). Then it checks the number of employees required by v, and if it is more than one, it iterates the remaining candidate allocations by always removing the first one and including it in the solution structure (*sol*) (lines 7 to 14). If at any iteration there are not enough candidates for the required number of employees, then v is unassigned from all its previous allocations since an activity cannot be assigned partially to less than the number of employees required. Finally, once v has been assigned, the GetRelated procedure searches for the related activities of v. Related activities are those involved in a time-dependent constraints with v. The list of related activities *lrv* is returned by the procedure.

The ProcessDep method assigns activities in the *lrv* in a similar way as Process. In addition, it enforces any time-dependent constraints by restricting the start time of dependent activities using the function ConsiderRC .

Algorithm 4 Greedy Heuristic 3 (GH3): Process and ProcessDep methods

```
 1: procedure Process(v)
 2:     can ← AllocPossibleAny(v,sol)
 3:     Sort(candidates)
 4:     if can  is not empty then
 5:         c ← can.remove(0)
 6:         Include(c,sol)
 7:         i ← v.required
 8:         for  i > 1 do
 9:             if can  is not empty then
10:                 c ← can.remove(0)
11:                 Include(c,sol)
12:             else
13:                 Unallocate(v,sol)
14:     lrv ← GetRelated(v)
15:     return lrv
```

```
 1: procedure ProcessDep(v, v2, lrv)
 2:     can ← AllocPossibleAny(v2,sol)
 3:     ConsiderRC(v, v2, can)
 4:     if can  is not empty then
 5:         Sort(can)
 6:         c ← can.remove(0)
 7:         Include(c,sol)
 8:         i ← v.required
 9:         for  i > 1 do
10:             if can  is not empty then
11:                 c ← can.remove(0)
12:                 Include(c,sol)
13:             else
14:                 Unallocate(v,sol)
15:     return lrv
```

6.3.3.1 Functions for time-dependent activities constraints

The ConsiderRC procedure tries to assign a starting time to a dependent activity by using the starting time of the independent activity, which it relates to, and the rules imposed by the type of time-dependent constraint. There are five types: overlap, synchronisation, minimum difference, maximum difference and min-max difference. The procedure first identifies which type of constraint is required. The procedures are based on the insertion heuristics proposed by Solomon (1987) for the VRPTW. These are extended to address the time dependent constraints. Similar extensions have been proposed by Xu and Chiu (2001) for a field technician scheduling problem, and by Mankowska et al. (2014) for home health care routing. The novelty is that the proposed procedures can handle time windows and time-dependent activities at the same time.

Algorithm 5 Selecting Time-Dependent Function

```
1: procedure ConsiderRC(v, v2, can)
2:     for c ← 1, can do
3:         if RC(v, v2) = Sync then
4:             Sync(v, v2, can, can2, c)
5:         else if RC(v, v2) = Overlap then
6:             Overlap(v, v2, can, can2, c)
7:         else if RC(v, v2) = Min then
8:             Minimum(v, v2, can, can2, c)
9:         else if RC(v, v2) = Max then
10:            Maximum(v, v2, can, can2, c)
11:        else if RC(v, v2) = MinMax then
12:            MinMax(v, v2, can, can2, c)
13:    return can2
```

The overlap constraints validate that a new time window formed by the start time of the candidate allocation, plus the duration of the dependent activity $v2$, clashes with v, i.e. it overlaps time wise. If such clash exists, then there is an opportunity for both activities to overlap, and the candidate allocation is added to a second list of candidate allocations $can2$, which in addition to adhering to the time window of $v2$, can comply with the overlap constraint with v. If there is no clash, then the procedure uses the flexible time of the candidate allocation to delay its starting time as much as possible, i.e. $c.st = c.st + c.ft$. If the new delayed starting time exceeds the start time of v then it indicates that an overlap can be achieved by assigning some intermediate value. The candidate allocation is updated with a new delayed starting time and added to those candidates which comply with the overlap constraint ($can2$). For the pseudo-code of the function refer to the Algorithm 0 line 1. Figure 6.4 illustrates the handling of overlap type constraints.

Algorithm 6 Overlap

```
1: procedure Overlap(v, v2, can, can2, c)
2:     w2 ← NewW(c.st, c.st + v2.dur)
3:     cw ← Clash(v.win, w2)
4:     if cw is not nil then
5:         af = c.st + c.ft
6:         if cw.et ≤ af then
7:             c.ft = cw.et − c.st
8:         can2.add(c)
9:     else
10:        if (c.st + v2.dur) < v.st then
11:            tmp ← c.st + v2.dur + c.ft
12:            if tmp ≥ v.st then
13:                st = v.st − v2.dur
14:                dt = st − c.st
15:                ft = c.ft − dt
16:                fM = v.et − st
17:                if ft ≤ fM then
18:                    rft ← ft
19:                else
20:                    rft ← fM
21:                it = c.it + dt
22:                c.st ← st
23:                c.ft ← rft
24:                c.it ← it
25:                can2.add(c)
```

In the synchronisation type, the candidate allocation's start time ($c.st$) for $v2$ needs

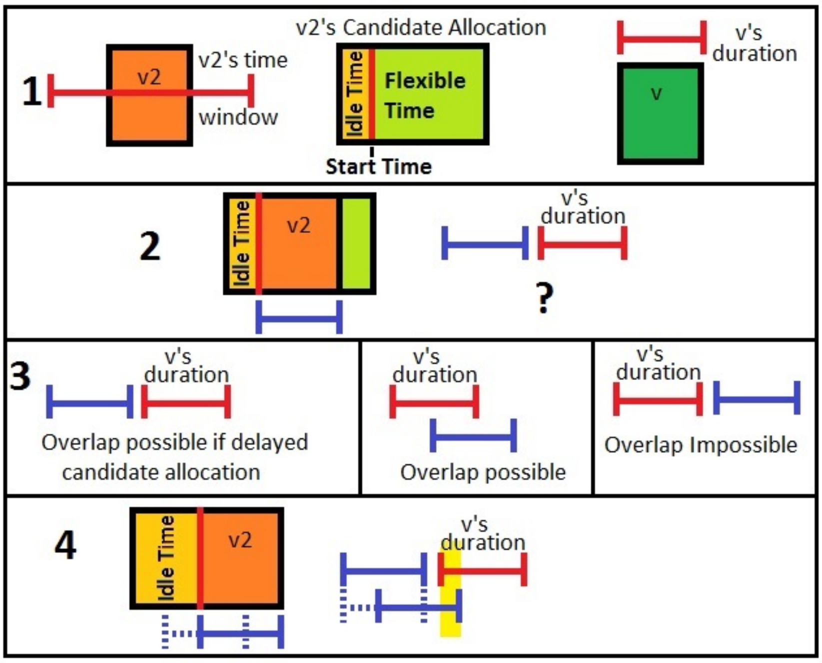

Figure 6.4: Test of time-dependent type overlap between a candidate allocation for $v2$ and independent visit v. The pseudo-code is in Algorithm 0

to be the same time as v's start time ($v.st$). Therefore, the procedure verifies if $v.st$ is equal to $c.st$. If the values are not the same, then it considers using the flexible time of the candidate allocation ($c.ft$) to make $c.st$ the same as $v.st$. Using the flexible time, a maximum start time mst for the candidate allocation is obtained. Next, it is verified if $v.st$ is between the two values $c.st$ and mst. If it is, then a new $c.st$ is set accordingly to enforce the synchronisation constraint. For the pseudo-code of the function refer to Algorithm 0 line 1. Figure 6.5 depicts the steps previously defined.

Algorithm 7 Synchronisation

1: **procedure** SYNC($v, v2, can, can2, c$)
2: $\quad mSt \leftarrow c.st + c.ft$
3: $\quad$ **if** $c.st \leq v.st$ AND $v.st \leq mst$ **then**
4: $\quad\quad st = v.st$
5: $\quad\quad dt = v.st - c.st$
6: $\quad\quad ft = 0$
7: $\quad\quad it = c.it + dt$
8: $\quad\quad c \leftarrow$ NEWCA(st, it, ft, emp)
9: $\quad\quad can2.add(c)$

In the minimum difference constraint type, $v2$ is required to start at least rcV time units after the commencement of v. With a shift similar to the synchronisation type, the function tests whether the candidate allocation's start time is greater than v's start time plus the time units (rcV), i.e. $c.st \geq v.st + rcV$. If it is greater, then the current candidate allocation is a valid one, and is added to the list $can2$. If it is not greater, the possibility of using the candidate allocation's flexible time by testing if $c.st + c.ft \geq v.st + rcV$ is considered. If the latest test is true, then there are some values of $c.st$ that can comply to the minimum difference constraint. The

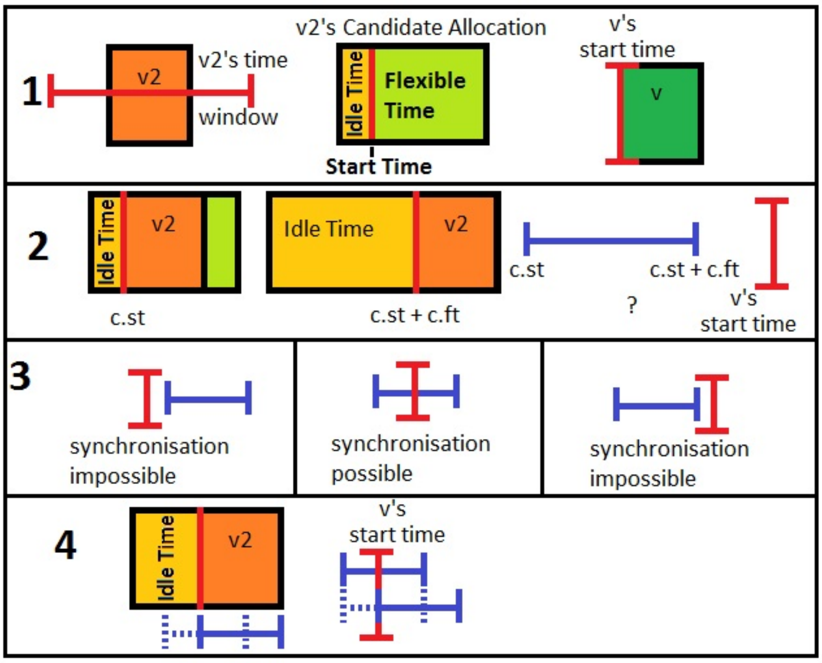

Figure 6.5: Test of time-dependent type synchronisation between a candidate allocation for $v2$ and independent visit v. The pseudo-code is in Algorithm 0

candidate allocation structure is updated accordingly to a new $c.st$ with flexible time reduced. The reader is referred to Algorithm 0 line 1 for the pseudo-code version of the procedure. Figure 6.6 demonstrates how the minimum difference function operates.

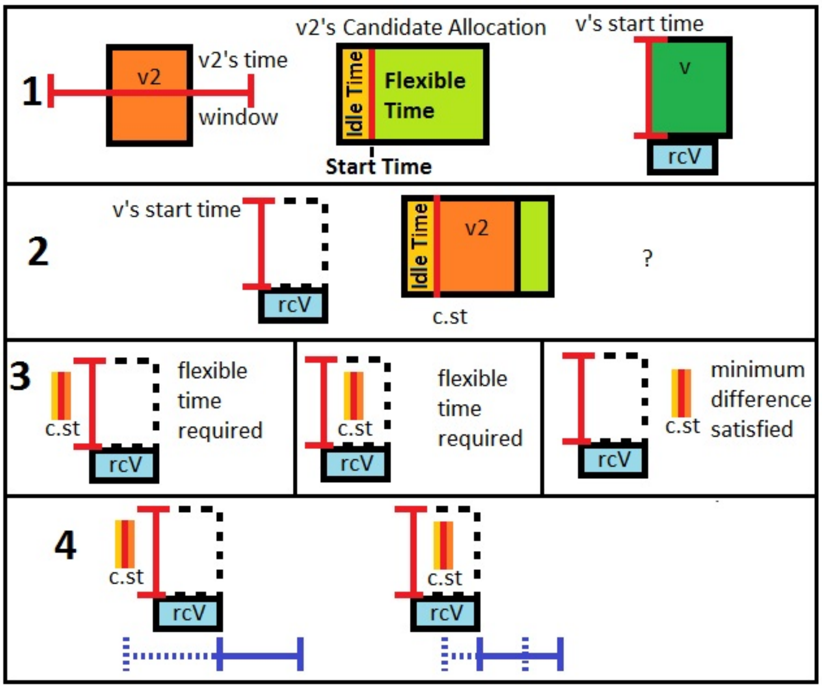

Figure 6.6: Test of time-dependent type minimum difference between a candidate allocation for $v2$ and independent visit v. The pseudo-code is in Algorithm 0

The maximum difference type requires $v2$'s start time to fall between the start time

Algorithm 8 Minimum Time Difference

```
1: procedure Minimum(v, v2, can, can2, c)
2:     rcV ← getRCVal()
3:     if v2 is dependat then
4:         sft ← c.st + c.ft
5:         if c.st ≥ v.st + rcV then
6:             can2.add(c)
7:         else if sft ≥ v.st + rcV then
8:             dt ← sft − (v.st + rcV)
9:             st ← (v.st + rcV)
10:            c.ft ← dt
11:            c.it ← (c.it + v.st + rcV − c.st)
12:            c.st ← st
13:            can2.add(c)
14:    else
15:        if (c.st + rcV) ≤ v.st then
16:            dt ← (v.st − rcV) − c.st
17:            if dt < c.ft then
18:                c.ft ← dt
19:                c.it ← c.it + dt
20:            can2.add(c)
```

of v and at most a new deadline defined by the addition of some time units rcV, i.e. $v.st + rcV$. If the candidate allocation's start time is not between the two values, then it is only appropriate to attempt to delay $c.st$ by using the flexible time $c.ft$ if $c.st$ is less than $v.st$. If the candidate allocation's starting time ($c.st$) is already after $v.st + rcV$ then that candidate allocation becomes invalid as the shifting with flexible time only allows delays. In this case, shifting forward the independent visit could work, but that depends on whether such activity can be delayed as it is assumed that it has already been set and it might be involved in other time-dependent constraints or on the limit of its time window. The pseudo-code for this function is available in Algorithm 0 line 1. Figure 6.7 illustrates the maximum difference function procedure.

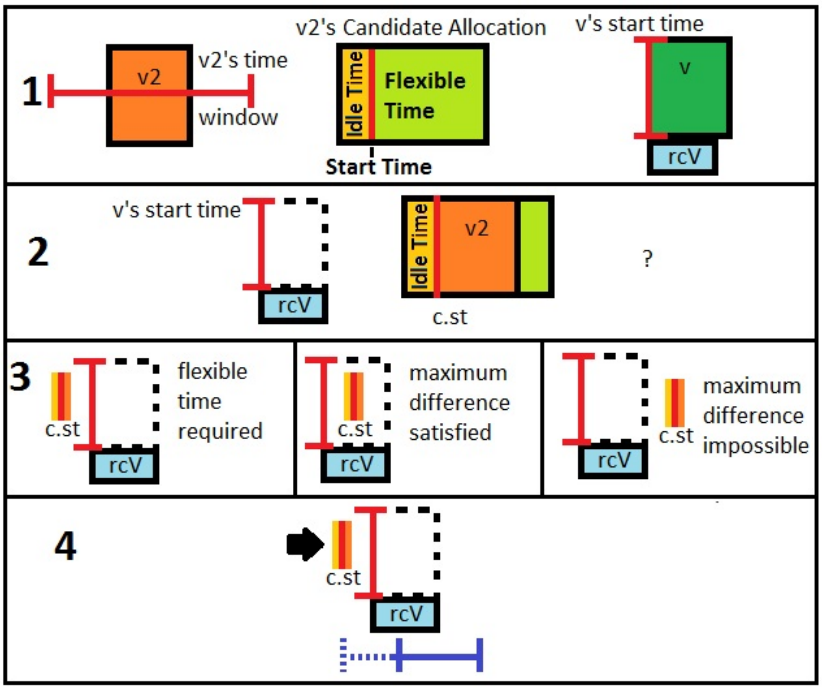

Figure 6.7: Test of time-dependent type maximum difference between a candidate allocation for $v2$ and independent visit v. The pseudo-code is in Algorithm 0

Algorithm 9 Maximum Time Difference

```
procedure MAXIMUM(v, v2, can, can2, c)
    rcV ← GETRCVAL()
    if v2 is dependat then
        sft ← c.st + c.ft
        cn1 ← v.st ≤ c.st
        cn2 ← c.st ≤ (v.st + rcV)
        if AND(cn1, cn2) then
            sft ← c.st + c.ft
            if sft ≥ (v.st + rcV) then
                c.ft ← sft − (v.st + rcV)
            can2.add(c)
        else if v.st ≤ sft then
            st ← c.st
            dt ← c.st + c.ft − v.st
            c.st ← v.st
            if sft ≤ v.st + rcV then
                c.ft ← dt
            else
                c.ft ← rcV
            c.it ← c.it + c.st − st
            can2.add(c)
    else
        sft ← c.st + c.ft
        cn1 ← c.st ≤ v.st
        cn2 ← v.st ≤ (c.st + rcV)
        if AND(cn1, cn2) then
            if v.st ≤ sft then
                c.ft ← c.ft − (sft − v.st)
            can2.add(c)
        else
            t1 ← c.st > v.st
            tmp ← c.st + c.ft + rcV
            if ¬(t1 AND tmp < v.st) then
                st ← c.st
                c.st ← v.st − rcV
                if sft ≤ v.st then
                    dt ← sft − (v.st − rcV)
                    c.ft ← dt
                    c.it ← c.it + c.st − st
                    can2.add(c)
                else
                    c.ft ← rcV
                    c.it ← c.it + c.st − st
                    can2.add(c)
```

The combined type minimum-maximum (min-max) difference can be seen as an additional time window imposed on $v2$ which depends on the v's start time. The new imposed time window defines a minimum starting time of $v.st + rcV1$ and a maximum starting time of $v.st + rcV2$. It is assumed that $rcV1 \leq rcV2$. Similarly, with the maximum difference type, and due to the restriction on the shifting movement in the candidate allocation's starting time, only cases where $c.st \leq v.st + rcV1$ are considered, when $c.st$ does not comply with the new imposed time window. If $c.st$ complies with $(v.st+rcV1) \leq c.st \leq (v.st+rcV2)$ then the constraint is enforced by the current configuration. If $c.st < (v.st + rcV1)$ and then there is flexible time to consider, a delay of $c.st$ may enforce the constraint. The pseudo-code of this function is available in appendix 0 line 1. Figure 6.8 shows the procedure of min-max described in this paragraph.

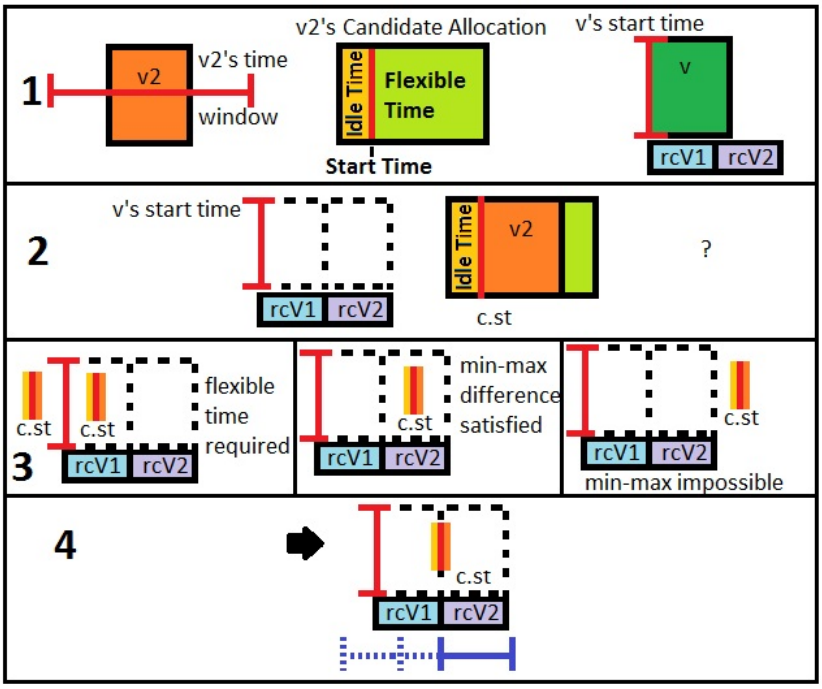

Figure 6.8: Test of time-dependent type minimum-maximum difference between a candidate allocation for $v2$ and independent visit v. The pseudo-code is in Algorithm 0

6.3.4 Introducing a summary of candidate allocations: GH4

The next improvement GH4 focuses on the available candidate allocations that have comply with both time window and time-dependent constraints. If there are more candidate allocations than number of employees required for the activity to be assigned ($v2$), the algorithm determines which candidate allocation to choose. Another factor to consider is that not all candidate allocations cover the same time frame, and in the case when more than one employee is required, the candidate allocations need to

Algorithm 10 Minimum-Maximum Time Difference

```
procedure MinMax(v, v2, can, can2, c)
    rcV1 ← getRCVal(min)
    rcV2 ← getRCVal(max)
    if v2 is dependat then
        mn ← v.st + rcV1
        mx ← v.st + rcV2
        sft ← c.st + c.ft
        cn1 ← mn < c.st
        cn2 ← c.st ≤ mx
        if AND(cn1,cn2) then
            if sft ≥ mx then
                c.ft ← c.ft − (sft − mx)
            can2.add(c)
        else if mn ≤ sft then
            st ← c.st
            dt ← sft − (v.st + mn)
            c.st ← mn
            if mn + dt ≥ mx then
                c.ft ← mx − mn
            else
                c.ft ← dt
            c.it ← c.it + c.st − st
            can2.add(c)
    else
        mn ← c.st + rcV1
        mx ← c.st + rcV2
        if ¬(mx < v.st) then
            dt ← v.st − mx
            c.st ← c.st + dt
            tmp ← mx − mn
            if (mx − v.st) < tmp then
                c.ft ← mx − v.st
            else
                c.ft ← mx − mn
            c.it ← dt
            can2.add(c)
```

be synchronised. As a result, the improvement is to create a catalogue of candidate allocations covering different time frames and enable the algorithm to choose a time frame that has all employees required. Figure 6.9 illustrates a collection of candidate allocations for $v2$. The candidate allocations (A, B, C, D and E) satisfy $v2$'s time window constraint. If $v2$ requires only one employee, any of the candidates could be an appropriate selection. However, if $v2$ requires two employees, the only possible combinations are the following pairs: A & C, A & E, C & E, and D & E with an adequate starting time that matches both of the selected candidates in each pair. An impossible combination, for example, is A & B, the main reason being that both candidate allocations are for the same employee (Employee 1) and $v2$ requires two different ones. The pair also clearly does not overlap in time. Another impossible combination is B & C since both do not overlap at any time so as to start $v2$ at the same time. The resulting catalogue structure captures in a better way all the possible options for allocations, allowing a better selection of an appropriate $v2$'s start time.

The catalogue structure shown at the bottom of Figure 6.9 is created as follows: every candidate allocation provides the start time and end time in which it can hold the activity. For example, in the Figure 6.9 candidate allocation, A provides 12:00 as a start time and 14:10 as the limit. All such time values are collected in a list (*dcList*) and sorted in ascending manner. In the figure the blue lines show the limits of every candidate allocation and provide the time confirming the list *dcList*. Then every candidate allocation that covers a time frame defined by consecutive nodes in the list *dcList* are added to the node by a reference. Such additions mean that for the time defined by each of the nodes (12:00, 12:10, 14:00, etc.) in the list, a collection of candidate allocations is kept forming the catalogue. Once the catalogue is formed, it is easier to identify, if for example three employees are required for $v2$, that the only available time where it is possible is from 14:00 (inclusive) until 14:10 (exclusive). The pseudo-code for the creation of the catalogue structure can be found in appendix D.2.

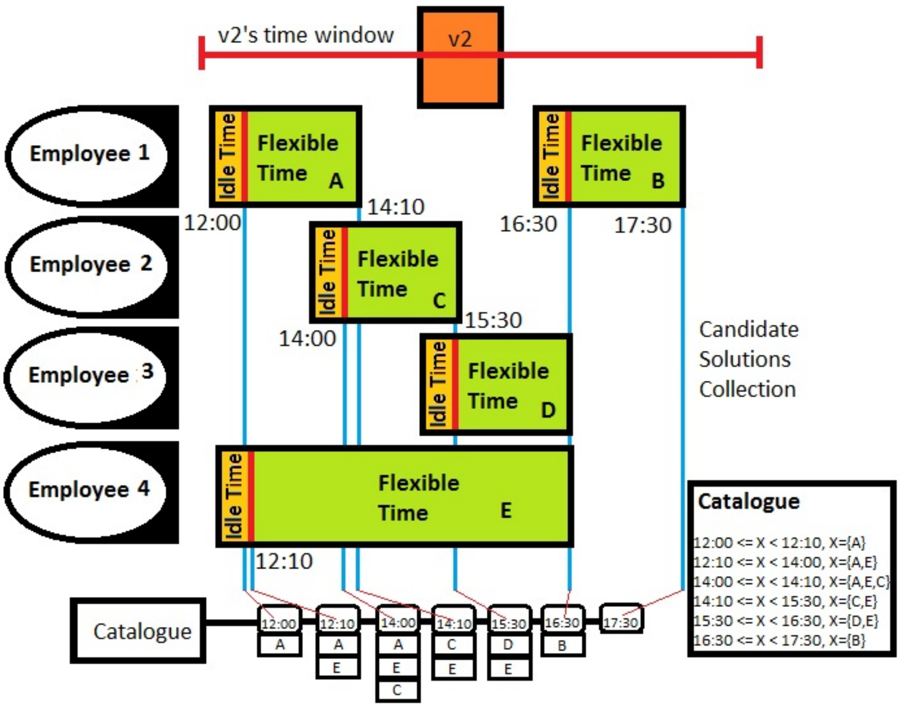

Figure 6.9: A catalogue structure creation from a collection of candidate allocations for visit $v2$.

The updated version of the procedures PROCESS and PROCESSDEP, that includes the construction of the catalogue to choose which candidate allocations are assigned to the solution, is shown in pseudo-code 11.

The statements after the formation of the Catalogue structure (line 4 in PROCESS and line 6 in PROCESSDEP), choose the last index that has enough candidate allocations as employees required and continues with the heuristic. The statement that follows verifies if there is index with that requirement (lines 5 and 6 respectively).

In the following section the remaining candidate allocations that could provide alternative solutions are explored to some degree.

Algorithm 11 Greedy Heuristic 4 (GH4) with call to function CATALOGUE to index candidate allocations.

```
procedure PROCESS(v)
    cand ← ALLOCPOSSIBLEANY(v,sol)
    cover ← CATALOGUE(can)
    can ← cover.getLast()
    if can is not empty then
        c ← can.remove(0)
        INCLUDE(c,sol)
        i ← v.required
        for i > 1 do
            if can is not empty then
                c ← cand.remove(0)
                INCLUDE(c,sol)
            else
                UNALLOCATE(v,sol)
    lrv ← GETRELATED(v)
    return lrv
```

```
procedure PROCESSDEP(v, v2, lrv)
    can ← ALLOCPOSSIBLEANY(v2,sol)
    CONSIDERRC(v, v2, can)
    if can is not empty then
        cover ← CATALOGUE(can)
        can ← cover.getLast()
        c ← can.remove(0)
        INCLUDE(c,sol)
        i ← v2.required
        for i > 1 do
            if can is not empty then
                c ← can.remove(0)
                INCLUDE(c,sol)
            else
                UNALLOCATE(v,sol)
    return lrv
```

6.3.5 Branching: GH5

The final improvement (GH5) to the greedy heuristic considers multiple options when choosing a node from the catalogue structure. All nodes within the catalogue hold possible configurations of a visit to be assigned to the solution structure. Previously (in GH4), only one is chosen but any of them could work if containing enough candidate allocations for the number of employees required. If the solution structure is copied and each copy is given a different option from the catalogue, many more options can be analysed. This branching procedure allows GH5 to analyse mxB options where $mxB \leq catalogue.size$. After analysing the options the best improvement (minimisation) in the objective function can be chosen. The best one is defined by the option that reduces the minimisation objective function value the most. Such a branching procedure can be seen as performing local searches with every activity to a certain extent.

Branching the solution structure requires cloning so that each clone can be evaluated with the different options. Once the best one is obtained, the clones can be discarded and the next iteration for assignment of an activity commences. Or each option can continue the search process independently in the hope that compromising on choosing the best one earlier in the search process might result in a better final result as the search progresses. The only problem with such an approach is that for large size scenarios the memory requirements of the computer are rapidly filled.

Including this sort of branching requires the addition of another parameter to GH5 mentioned earlier, the *maxBranching* (mxB). This parameter limits the possibilities that are analysed during the branching. In other words, it limits the size of the neighbourhood. If *maxBranching* is bigger than the possibilities of assignment (catalogue nodes) then all possibilities are considered. But, if *maxBranching* is less than the number of available nodes in the catalogue then only *maxBranching* nodes are considered. The choice of which nodes are considered is a random one. The pseudo-random generator is fed with the same seed so as to maintain deterministic results when repeating experiments, but the seed can be changed if required.

Allowing the evaluation of a neighbourhood of possible allocations changes the pseudo-code of the heuristic. The final pseudo-code presenting all the different additions to the original bin-packing inspired heuristic is presented in Algorithm 12. Similarly updates to PROCESS and PROCESSDEP are presented in Algorithm 13.

Algorithm 12 Greedy Heuristic 5 (GH5)

```
procedure SOLVE
    visitList ← copy of visits(V)
    sol ← NEWSOLUTIONSTRUCTURE
    SORT(visitList, listCriterion)
    while visitList is not empty do
        SORT(sol, solCriterion)
        INITIALISE(solToEval)
        v ← visitList.remove(0)
        solutions ← PROCESS(v)
        for sc ← 1, solutions do
            l ← sc.lrv
            if l.size > 0 then
                while ¬EMPTY(l) do
                    v2 ←GET(l,0)
                    S ← PROCESSDEP(v, v2, l)
                    visitList.remove(v2)
                    solToEval.addAll(S)
            else
                solToEval.add(sc)
        bestSol ← MAX
        best ← −1
        for sc ← 1, solToEval do
            tmp ← CALCOBJFUN(sc.sol)
            if bestSol > tmp then
                bestSol = tmp
                best ← sc
        if best¬ − 1 then
            sol ← best.sol
```

6.3.6 Optional Improvement: Multi-start

As discussed earlier in subsections 6.3.1.1 and 6.3.1.2, the parameters *solCriterion* and *listCriterion* determine the order in which employees and activities are evaluated in the main cycle of GH. Other factors, discussed in previous sections, alter the next activity in line for assignment by GH. One of the factors is if an activity has time-dependent constraints then all the related activities are processed immediately after the independent one and then removed from the *visitList*. Another factor is if an employee has neither the skills nor the time to perform an activity the next employee is evaluated, "next" means the following node in the solution structure *sol*. Such move-

Algorithm 13 Greedy Heuristic 5 (GH5): PROCESS and PROCESSDEP

```
procedure PROCESS(v)
    cand ← ALLOCPOSSIBLEANY(v,sol)
    cover ← CANTIMEINDEX(can)
    for dc ← 1, cover do
        if v.required > dc.eCover then
            cover.remove(dc)
    if coveris not empty then
        solutions
        if cover.size > MxPar then
            rem ← cover.size − MxPar
            for r ← 1, rem do
                i ←RAND(cover.size)
                cover.remove(i)
        for dc ← 1, cover do
            scenario ← CLONE()
            solutions.add(scenario)
        for i ← 1, solutions.size do
            s ← solutions.get(i)
            can ← cover.get(i).can
            c ← can.remove(0)
            INCLUDE(c,sc.sol)
            i ← v.required
            for  i > 1 do
                if ¬EMPTY(can) then
                    c ← cand.remove(0)
                    INCLUDE(c,s.sol)
                else
                    UNALLOCATE(v,s.sol)
            s.lrv ← GETRELATED(v)
    return solutions
```

```
procedure PROCESSDEP(v, v2, lrv)
    can ← ALLOCPOSSIBLEANY(v2,sol)
    CONSIDERRC(v, v2, can)
    if can  is not empty then
        cover ← CANTIMEINDEX(can)
        for dc ← 1, cover do
            if v.required > dc.eCover then
                cover.remove(dc)
        if coveris not empty then
            solutions
            if cover.size > MxPar then
                rem ← cover.size − MxPar
                for r ← 1, rem do
                    i ←RAND(cover.size)
                    cover.remove(i)
            for dc ← 1, cover do
                scenario ← CLONE()
                solutions.add(scenario)
            for i ← 1, solutions.size do
                s ← solutions.get(i)
                can ← cover.get(i).can
                c ← can.remove(0)
                INCLUDE(c,s.sol)
                i ← v2.required
                for  i > 1 do
                    if ¬EMPTY(can) then
                        c ← cand.remove(0)
                        INCLUDE(c,s.sol)
                    else
                        UNALLOCATE(v,s.sol)
                s.lrv ← GETRELATED(v)
        return solutions
```

ment is likely to be overridden if any of AllocPossibleAny, ConsiderRC and Catalogue functions are used, as these functions/procedures designate a collection of possible employees in one way or another. The running time of GH1-GH4 versions of the greedy heuristic (running times discussed in more detail in the Results section 6.4) are within milliseconds and for GH5 within seconds. The advantage of such short running time when compared to the two hours given to the mathematical solver, is that any version of the greedy heuristic could be restarted and run with different initialisation parameters. Therefore, additional improvements could be made if allowing the heuristic to re-start with different values for *solCriterion* and *listCriterion* and retaining only the best one. Such approach is discussed in the experiments in Section 6.4. Moreover, a multi-start approach could take advantage of a cluster environment or multiple processor units.

6.4 Experimental Results

The greedy heuristic (GH) has been developed in Java. Five different versions of the greedy heuristic are analysed (GH1 - GH5). Every version matches one of the improvements mentioned in section 6.3: GH1 refers to the original bin-packing inspired heuristic; GH2 introduces the broadening of search for inter idle time; GH3 incorporates the functions tailored to tackle time-dependent activities constraints; GH4 introduces the catalogue of candidate allocations; and, GH5 includes local search for a limited number of possible solutions, i.e. temporary branching.

Regarding the parameters of the GH, *solCriterion* and *listCriterion*, every combination is tested and the best result provided. For GH5 the additional parameter *maxBranching* is evaluated with four different values: 10, 20, 40, 50. The values represent the maximum number of candidate solutions that will be considered.

Every version of GH is compared to the benchmark results obtained by the mathematical solver using the MILP model from chapter 5.3. The summarised results for every version of GH are presented in this chapter using a table with descriptive statistics and a graphical representation. Both the tables and graphs seek to illustrate the observations and findings. However, for the individual instance result of each of the 375 instances using all versions of GH, the reader is referred to Appendices D.3 to D.8 for closer scrutiny if desired.

The gap reported in this section is calculated according to equation 6.1. The principle is the following, because it is a minimisation problem, in every comparison between the solver and a version of the greedy heuristic the best result(smallest objective function value) is subtracted from the worst result (largest objective function value) and normalised according to the best result. Absolute value of the denominator is necessary to avoid providing negative gaps.

$$gap = (Worst - Best)/ABS(Best) \tag{6.1}$$

6.4.1 GH1 Results

Table D.1 (see Appendix D.3) provides the value of the objective function obtained using GH1 and the running time in milliseconds. If the solution obtained is infeasible due to constraint violations, the instances are reported as "-".

The results are compared against those obtained by the mathematical solver (reported in chapter 5.3). GH1 obtains better results based on the value of the objective function (equation 5.12) for 124 instances out of the 375, whereas the mathematical solver obtains better results for 173. It also finds solutions for 78 instances but such solutions are infeasible due to violation in the time-dependent constraints, as GH1 does not guarantee complying with them.

6.4.1.1 Results group classification

In order to provide a comparison of GH1 against the solver results, it is necessary to classify the results. Six groups are created: group 1 where the solver obtains an optimal solution; group 2 where the solution obtained by the solver is better than the solution obtained by GH1, but such solution is not optimal; group 3 where GH1 is better than the solver results. Therefore, groups 1, 2, 3 have results for both methods GH1 and solver. Group 4 is where only GH1 results are available; group 5 is where only solver values are available and group 6 is where neither the solver nor GH1 could obtain a feasible result. The proposed group division allows three comparisons. The first one compares how good GH1 is against known optimal values. The second is how far each method is from the other where the optimal value is unknown. Finally, it compares the number of instances for which GH1 can find a solution but the solver is unable to do so, and vice versa.

Using the proposed group classification, Table 6.1 shows the number of instances in each group. It is known from the results obtained by the MILP model (see Chapter

5.3) that there are 34 instances with known optimal values. However, Group 1 in this classification only contains 23 instances. The 11 missing instances are not in group 1 because they are part of group 5, as there is no result provided by GH1 for them. Table 6.1 also shows descriptive statistics regarding the gap between the value obtained by the solver and by GH1.

The gap value can only be calculated for groups where both results are available, i.e. groups 1, 2, 3. The gap calculation for groups 1 and 2 is $(GH1-Solver)/ABS(Solver)$ and for group 3 is $(Solver-GH1)/ABS(GH1)$ based on the previously defined function 6.1.

Group	# Instances	Min	Q1	Median	Q3	Max	Mean	Std Dev
Group 1	23	30.43%	69.19%	396.93%	1502.52%	70683.53%	5703.13%	16719.20%
Group 2	150	0.78%	37.22%	156.13%	798.15%	20171.73%	717.62%	1886.85%
Group 3	93	1.45%	27.85%	59.55%	82.04%	146.77%	58.80%	34.26%
Group 4	31							
Group 5	72							
Group 6	6							

Table 6.1: Summary of GH1 result values compared to the mathematical solver. The number of instances in each group is shown. Groups are according to definition in 6.4.1.1. The minimum, first quartile, median, third quartile, maximum, mean and standard deviation of the gap are shown for groups 1, 2 and 3. Groups 4, 5 and 6 do not have values for both methods (GH2 or solver) so the gap cannot be obtained.

Figure 6.10 displays the gap as appropriate for groups 1, 2 and 3. It is observed that for Group 1 the best gap achieved by GH1 is of 30% from the optimum (see row 1 in Table 6.1). The best gap achieved in Group 2 is less than 1% but the mean is of 700%. Overall, the solver values when compared to GH1 (Group 3) are closer to GH1 values than the values of GH1 when compared to the solver (Group 2), which suggests that the solver is not far from the value obtained by GH1 when GH1 is better but GH1 is not so close to the solver results when the solver is better.

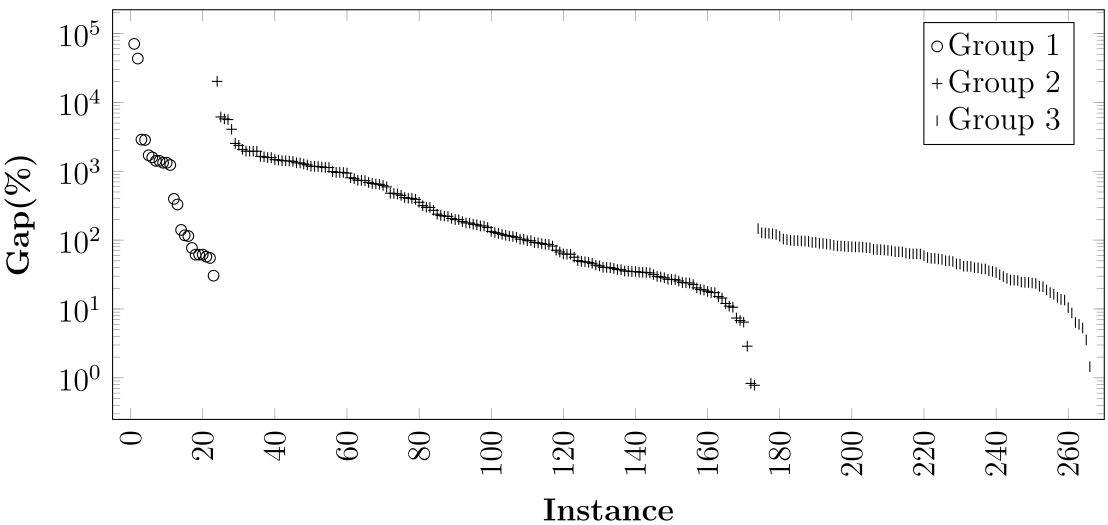

Figure 6.10: Computed gaps for groups 1, 2 and 3 when using GH1

6.4.2 GH2 Results

In this section the results obtained by GH2 are presented. Such results are also compared to the solver ones, in a similar way as in section 6.4.1. The main difference between GH1 and GH2 is that GH2 explores any available idle times within the working time of employees, in comparison to GH1 which only attemps to assign them after the last performed activity. Therefore, the search space available to GH2 is bigger than GH1, which might result in a better outcome.

The GH2 version finds better results than the solver for 92 instances. The results obtained for 131 instances are infeasible due to the violations of time-dependent constraints. GH2 does not handle time-dependent constraints.

6.4.2.1 Results group classification

The results are classified into six groups as defined in the previous section. Group 1 contains optimal solutions that are known and GH2 has a feasible comparable value. Group 2 includes the non-optimal solver results that are better than the ones found by GH2. Group 3 has results where GH2 is better. Groups 1, 2, 3 assume there are results for both methods in order to allow the comparison. Group 4 contains instances where only GH2 results are available. Group 5 includes instances where only the solver reports feasible results. And, in group 6 neither GH2 nor the solver find feasible results. Table 6.2 contains the number of instances in each group and information about the gap between the values obtained by GH2 and the solver. The gaps are computed as defined in 6.4.1.1.

Group	# Instances	Min	Q1	Median	Q3	Max	Mean	Std Dev
Group 1	15	30.43%	61.48%	108.79%	138.22%	28673.03%	2360.64%	7343.83%
Group 2	116	0.78%	35.54%	95.41%	391.74%	11036.11%	499.07%	1287.68%
Group 3	76	1.30%	26.62%	66.01%	88.04%	146.77%	60.57%	36.15%
Group 4	16							
Group 5	131							
Group 6	21							

Table 6.2: Summary of GH2 result values compared to the mathematical solver. The number of instances in each group is shown. Groups are according to the definition in 6.4.2.1. The minimum, first quartile, median, third quartile, maximum, mean and standard deviation of the gap are shown for groups 1, 2 and 3. Groups 4, 5 and 6 do not have values for both methods (GH2 or solver) so the gap cannot be obtained.

Only considering the number of instances in Groups (3 and 4) it can be said that GH2 is worse than GH1 since GH1 has 124 instances with better results than the solver compared to 92 of GH2. However, there is a rise in the number of infeasible solutions obtained by GH2 (131) compared to GH1 (78). In total, 52 more instances with infeasible solution are presented in GH2. It could be argued that expanding the search space available to the greedy heuristic without addressing time-dependent constraints does not improve the number of instances with a better objective value. However, the gap in groups 1 and 2 for GH2 decreases in comparison to GH1, suggesting that even though the number of instances with better results for GH2 is less than GH1, their quality is better (mean of group 1 in Table 6.2 is smaller than the one of group 1 in Table 6.1). The solver gaps (Group 3) increase from the second quartile onwards with respect to GH1. Such an increase also indicates that the quality of solutions achieved by GH2 increases. Figure 6.11 displays the gap for groups 1, 2 and 3. Table D.2 (see Appendix D.4) contains the objective function value and computation time obtained by GH2 for all 375 instances, where time is given in milliseconds.

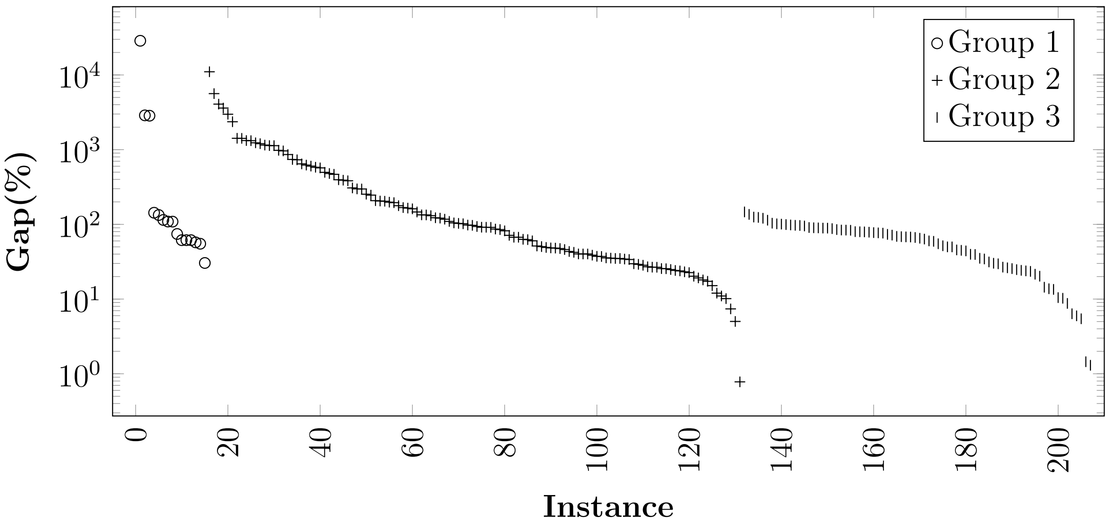

Figure 6.11: Computed gaps for groups 1, 2 and 3 when using GH2.

6.4.3 GH3 Results

This section presents the results of running version GH3 of the heuristic against the mathematical solver. GH3 is the first version that handles time-dependent constraints, hence it is able to find feasible results for all 375 instances. GH3 obtains better results for 178 instances out of the 375.

6.4.3.1 Results group classification

Results for GH3 are classified in four groups. Group 1 are instances for which an optimal solution is known. Group 2 are instances where the solver obtains better feasible solutions than GH3 although such solutions are not optimal. Group 3 have instances where GH3 obtains better results than the solver. And, Group 4 contains instances where results from GH3 are the only ones available, as the solver either runs out of memory or cannot load the instance due to its size. Table 6.3 shows the number of instances in each of the defined groups. In addition, Table 6.3 also contains the minimum, first quartile, median, third quartile, maximum, mean and standard deviation of the gap between the values of GH3 and the solver for groups for which the gap can be computed, i.e. groups 1, 2 and 3. The gap is computed as explained in section 6.4.1.1.

Group	# Instances	Min	Q1	Median	Q3	Max	Mean	Std Dev
Group 1	34	30.43%	65.28%	113.36%	233.38%	28673.03%	1517.42%	5262.85%
Group 2	163	0.71%	50.14%	125.26%	236.91%	9669.11%	339.49%	934.30%
Group 3	141	3.33%	40.30%	86.94%	147.90%	30039.00%	1478.82%	4806.81%
Group 4	37							

Table 6.3: Summary of GH3 result values compared to the mathematical solver. The number of instances in each group is shown. Groups are according to definition in 6.4.3.1. The minimum, first quartile, median, third quartile, maximum, mean and standard deviation value of the gap are shown for groups 1, 2 and 3. Group 4 has only results for GH3 so a gap value cannot be included.

The gap value for Group 1 decreases when compared to those obtained by GH2, which indicates that the quality of the solutions obtained by GH3, although still with a minimum of 30% overall, is increasing. In Group 2, quartiles Q1 and Q2 increase with respect to the values obtained by GH2. However, the third quartile Q3, the maximum gap and the mean are reduced. GH3 increases the number of instances with feasible solutions by 54 compared to GH1 and 86 compared to GH2. Interestingly, it is the solver gap to GH3 (Group 3) that increases overall, which indicates that the quality of solutions is improving. Figure 6.12 shows the gap values for Groups 1, 2 and 3.

Table D.3 (see Appendix D.5) includes the objective value obtain by GH3 for all 375 instances with the corresponding computation time in milliseconds. The average computational time for GH3 is 1.34 milliseconds with a standard deviation of 2.53.

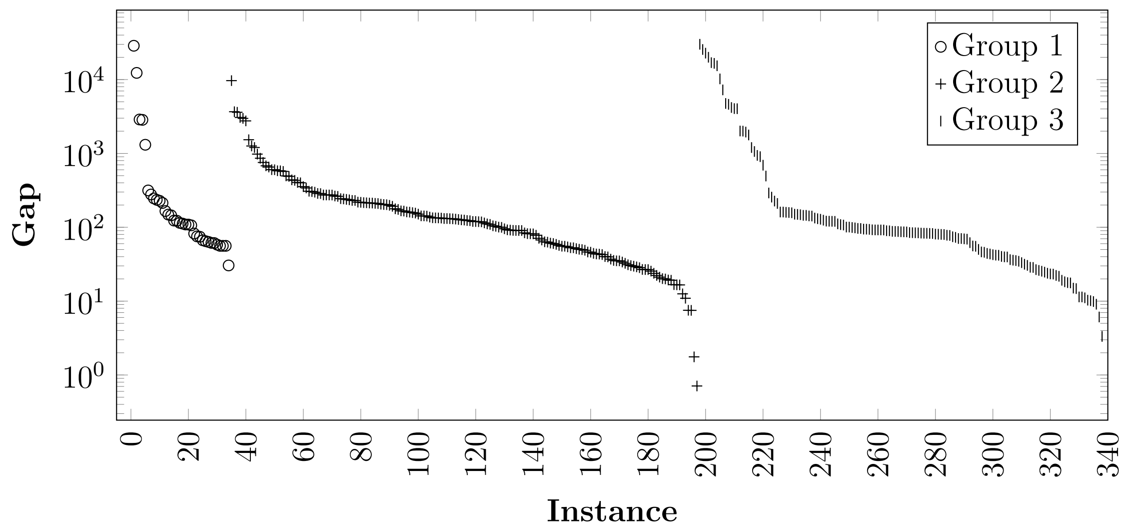

Figure 6.12: Gaps

6.4.4 GH4 Results

In this section results obtained by GH4 are presented. GH4 finds feasible solutions for all 375 instances, out of which 186 are better than the solver, almost 50%. The improvements in this version include the catalogue of candidate solutions which allows the heuristic to choose better in most cases. In terms of the number of instances with better results than the solver, it only finds 8 more compared to GH3.

6.4.4.1 Results group classification

The results are classified in a similar manner as with GH3. Group 1 contains instances with known optimal values obtained by the solver. Group 2 contains instances where the solver values are better than GH4. Group 3 includes instances for which GH4 obtains better results. Finally, Group 4 only has results for GH4. Table 6.4 provides information regarding the number of instances in each of the groups. Additionally it shows the minimum, first quartile, median, third quartile, maximum, average and standard deviation of the gap for Groups 1, 2 and 3.

Group	# Instances	Min	Q1	Median	Q3	Max	Mean	Std Dev
Group 1	34	5.35%	60.43%	75.06%	139.91%	20730.50%	1030.28%	3809.34%
Group 2	155	1.03%	32.87%	66.27%	121.26%	3028.72%	166.27%	407.69%
Group 3	149	0.91%	60.71%	100.72%	196.31%	27373.41%	2047.36%	5422.88%
Group 4	37							

Table 6.4: Summary of GH4 result values compared to the mathematical solver. The number of instances in each group is shown. Groups are according to definition in 6.4.4.1. The minimum, first quartile, median, third quartile, maximum, mean and standard deviation value of the gap are shown for Groups 1, 2 and 3. Group 4 only has results for GH4 so gap values cannot be included.

As GH4 finds feasible results for all instances, Group 1 remains with the same number of instances compared to GH3. What can be observed, however, is that the minimum gap obtained was reduced to 5%. Group 1 shows a reduced gap when compared to GH3. Eight instances are shifted from Group 2 to Group 3 compared to the results obtained by GH3. 50% of instances in Group 2 have a gap of 67% or less compared to the value of the solver. The gaps in Group 3 continue to increase as the solutions from the heuristic (GH4) continue to improve. Figure 6.13 shows the gap value for Groups 1, 2 and 3. Table D.4 (see Appendix D.6) shows the objective function value obtained using GH4 and the computation time in milliseconds. The average computational time increased to 2.21 milliseconds, even the maximum observed is still below 1 second, i.e. 85 milliseconds.

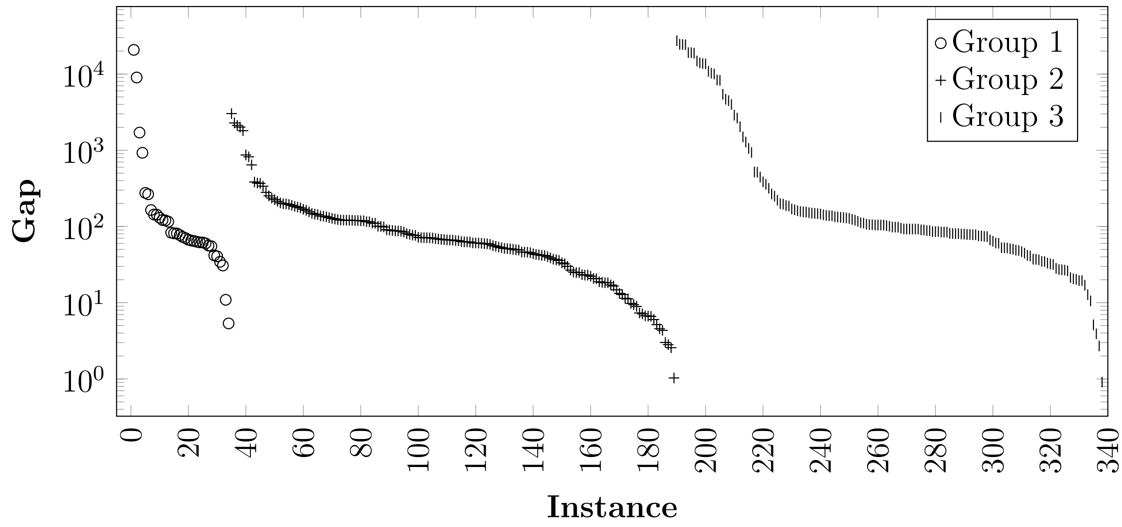

Figure 6.13: Gaps

6.4.5 GH5 Results

This section presents the results obtained by version GH5 of the greedy heuristic. This version, as with GH3 and GH4, obtains feasible results for all 375 instances. Moreover, it achieves better results than the mathematical solver for 203 instances out of the 375, which is 17 instances more than GH4. Therefore, it is the first version of the greedy heuristic that has more than 50% of instances with results better than the solver. All previous versions select the best candidate allocation for an activity to include in the solution structure, i.e. one level forward. This GH5 version evaluates two levels forward by cloning the solution structure to evaluate all candidate allocations options. GH5 then compares all the clones after an additional iteration and chooses the best value obtained so far. This is particularly important when assigning dependent activities, because they are constrained by the possibilities of their independent counterparts. GH5 also utilises a parameter which defines the number of candidate allocations that are considered during the two level forward evaluation. The results shown in this section correspond to the *maxBranching* parameter set to 10. In the next section the parameter is increased.

6.4.5.1 Results group classification

Results are classified using a similar approach as with GH3 and GH4, by forming four groups. Group 1 includes instances with known optimal values obtained by the mathematical solver. Group 2 contains instances where the solver value after 2 hours is better than GH5. Table 6.5 shows the number of instances in each group. It also shows the minimum, first quartile, median, third quartile, maximum, mean and standard deviation of the gap for groups 1, 2 and 3.

Group	# Instances	Min	Q1	Median	Q3	Max	Mean	Std Dev
Group 1	34	10.92%	39.23%	53.52%	73.56%	6323.37%	478.61%	1383.52%
Group 2	138	0.92%	20.88%	32.63%	58.81%	775.09%	55.00%	84.07%
Group 3	166	2.36%	74.26%	113.85%	398.10%	90936.22%	2387.09%	8656.99%
Group 4	37							

Table 6.5: Summary of GH5 result values compared to the mathematical solver. The number of instances in each group is shown. Groups are according to definition in 6.4.5.1. The minimum, first quartile, median, third quartile, maximum, mean and standard deviation value of the gap are shown for Groups 1, 2 and 3. Group 4 has only results for GH5 so gap values cannot be included.

Both mean gaps in Group 1 and 2 are reduced in comparison with GH4, which is a result of the improved quality of the solutions. As expected, the solver gaps grow in comparison to the heuristic. Figure 6.14 shows the gap value for Groups 1, 2 and 3. It is noticeable in the figure that Group 2 gaps concentrate at the bottom, 75% of the instances in this group achieve a gap of less than 60%. Table D.5 (see Appendix D.7) presents the objective value obtain with GH5, computational time used in milliseconds is also shown.

In reference to computational time, there is a significant increment compared to the average time of previous versions of the heuristic, from 2.21 milliseconds in GH4 to 1069 milliseconds (1 second) in GH5. Nevertheless, such time is still far from the two hours required by the solver. The maximum time spent in an instance was under 1 minute.

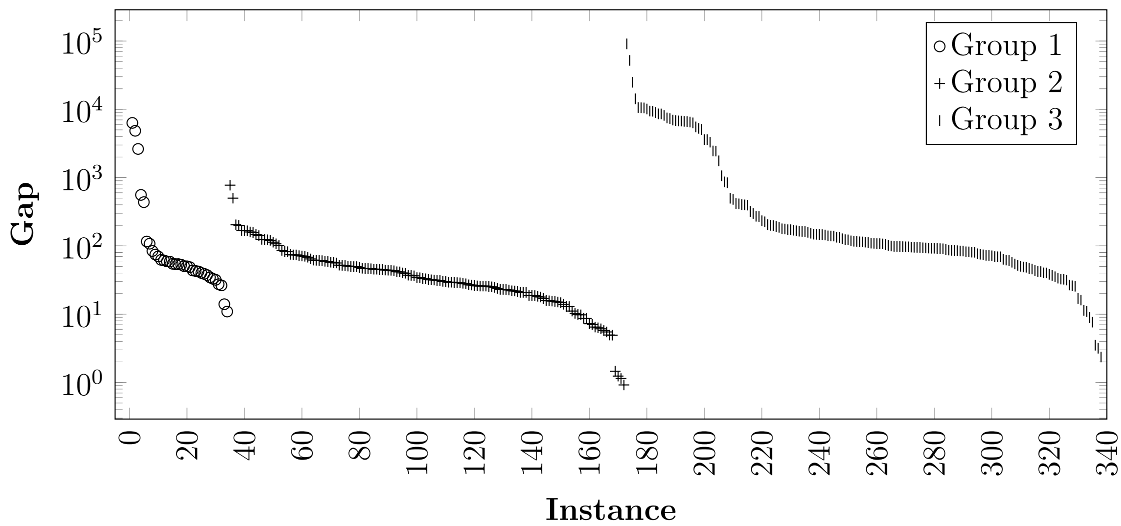

Figure 6.14: Gaps

6.4.6 GH5 Results with *maxBranching*= {20, 40 and 50}

In this section the results when increasing the parameter *maxBranching* are presented. In section 6.4.5 such parameter was set to 10. In this section results with values of 20, 40 and 50 are presented. Every increase aims to double the value of the parameter. However, when attempting to increase it to 80, some instances started running out of memory. As a result, 50 was chosen as the maximum parameter value used here that provides results for all instances. It is recognised that perhaps a different programming language or improvements on the implementation of the heuristic could prevent the out of memory error.

Table D.6 (see Appendix D.8) presents the objective function value obtained for each instance and the computational time in milliseconds for GH5 with *maxBranching* parameter set to 20, 40 and 50.

Table 6.6 provides a summary of the results obtained when using GH5 with *maxBranching* set to 20, 40, and 50. Group 1 across the different configurations of the *maxBranching* parameter seem to suggest that the mean of the gap remains approximately the same (478%). Similar results are presented in Group 2, where the mean does not change much across all configurations, in this case with a value of 53%. Regarding Group 3, it is observed that the configuration with the smaller gap on average is with *maxBranching* = 40.

Ver	G	#	Min	Q1	Median	Q3	Max	Mean	Std Dev
20	1	34	10.917%	39.227%	51.481%	73.561%	6323.371%	**478.252**%	1383.635%
40	1	34	10.917%	39.227%	51.481%	73.561%	6323.371%	**478.318**%	1383.615%
50	1	34	10.917%	39.227%	51.481%	**73.042%**	6323.371%	**478.297**%	1383.621%
20	2	137	0.921%	18.704%	**31.202**%	56.168%	775.090%	**53.478**%	83.981%
40	2	137	0.921%	18.704%	**31.644**%	56.168%	775.090%	**53.360**%	83.996%
50	2	137	0.921%	18.704%	31.644%	56.168%	775.090%	**53.337**%	84.002%
20	3	167	2.729%	75.835%	**113.212%**	387.628%	104351.684%	**2089.002**%	8612.728%
40	3	167	2.729%	75.054%	**111.908%**	387.628%	96239.209%	**2007.850**%	8008.893%
50	3	167	2.729%	75.054%	**111.908%**	387.628%	96239.209%	**2010.138**%	8010.609%

Table 6.6: Summary of GH5 result values compared to the mathematical solver. The number of instances in each group is shown. Groups are according to definition in 6.4.5.1. The minimum, first quartile, median, third quartile, maximum, mean and standard deviation value of the gap are shown for Groups 1, 2 and 3. Group 4 has only results for GH5 so gap values cannot be included.

Running time for GH5 depends on the *maxBranching* parameter. Table 6.7 shows the minimum, first quartile, median, third quartile, maximum, average and standard deviation of the computation time for each configuration. The minimum, Q1, median and Q3 values increase as the *maxBranching* increases. The maximum running time was observed when using GH5_40 (5138199, i.e. 85 min).

Version	Min	Q1	Median	Q3	Max	Mean	Std Dev
GH5(10)	0.00	9.00	55.00	337.50	54208.00	1069.70	5061.76
GH5(20)	0.00	10.00	60.00	710.00	164208.00	2318.52	12465.40
GH5(40)	0.00	10.00	71.00	1112.00	5138199.00	25262.96	305065.40
GH5(50)	0.00	9.00	74.00	1178.00	545000.00	5087.46	34360.10

Table 6.7: Running time of GH5 with different values for *maxBranching* parameter. Time is given in milliseconds.

6.4.7 Best overall results

In this section all results previously presented in this chapter are used to compare against the solver in order to obtain the minimum objective function value for each instance thus far. Out of the 375 instances, the solver obtains best results for 169. For the remaining 206 instances, a version of GH is better. The distribution of the best results across the different versions of GH is shown in Table 6.8. In many cases more than one version achieves the same result. If that is the case, the instances are counted for all versions that obtained the best objective value. According to the table GH5 with parameter *maxBranching* set to 40 obtains the best results. Contrary to the observation in section 6.4.6 regarding the average gap staying the same with different values of *maxBranching*, it is clear that increasing such parameter provides better quality results. The main reason why the gap mean seems unchanged is that the improvements are relatively small so they do not greatly affect the gap. It is also unexpected that setting *maxBranching* to 40 provides the almost the same number as using 50. Perhaps this is because the maximum number of candidate allocations is not always evaluated throughout the search. In many cases the configuration might allow 50 candidate allocations to be considered but during the search process such number might never be required or in rare occasions. It is also interesting that regardless of the value assigned to *maxBranching*, GH5 is clearly better than the previous versions (GH1 - GH4).

GH1	GH2	GH3	GH4	GH5(10)	GH5(20)	GH5(40)	GH5(50)
5	5	2	4	70	97	106	105

Table 6.8: Different versions of the greedy heuristic (GH1 - GH5_50) and the number of instances with the best result achieved.

6.5 Conclusion

In this chapter a greedy heuristic (GH) to tackle workforce scheduling and routing problems was presented. Five version of the heuristic were discussed. GH1 was inspired by the bin-packing problem. GH2 increased the search space but failed to find more instances with best result than GH1 for failure to support time-dependent constraints. As a result, it was necessary to create specialised procedures/functions to deal with the each type of time-dependent constraints that relate two or more activities. After the introduction of such functions, GH3 was able to find feasible solutions for all 375 instances. Further improvements are achieved when incorporating a catalogue structure of candidate allocations as options when assigning activities to the solution. GH4 was enabled to choose the best option by using the catalogue. The introduction of branching, i.e. multiple options, prove satisfactory as the quality of the results improved. The improvement was little in terms of the percentage of gaps achieved.

The use of specialised functions to tackle time-dependent constraints leads to better results. Figure 6.15 summarizes the types of moves that each function performs. Regarding computation time, the average in GH5 with *maxBranching* set to 50 takes just above 5 seconds. A single instance was reported to last up to 85 minutes in GH5 with *maxBranching* set to 40. The greedy heuristic in its different versions obtains better results than the solver for 206 out of the 375 instances (54%). GH5 with *maxBranching* set to 40 is the version that achieves the majority of best results 106.

In the next chapter the specialised functions are merged with some neighbourhood moves. The moves are integrated into a Tabu Search implementation seeking to achieve better results than those obtained by the Greedy Heuristic in its multiple versions.

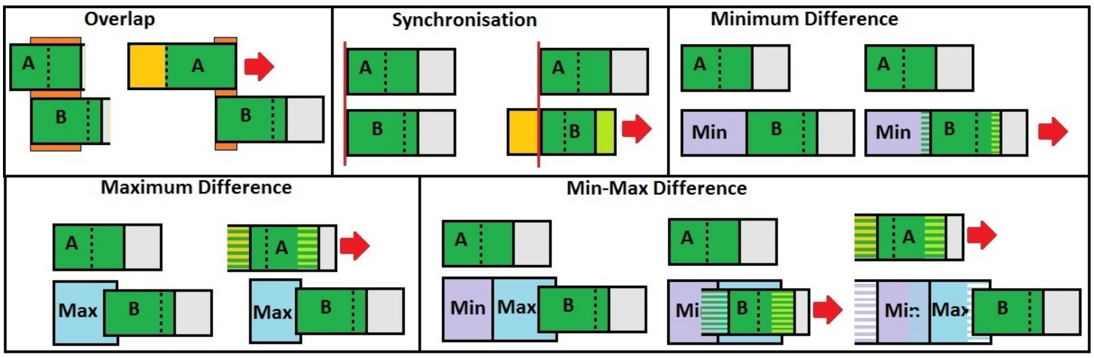

Figure 6.15: Examples of moves for every time-dependent constraint

Chapter 7

A Tabu Search Approach for WSRP

7.1 Introduction

In this chapter a Tabu Search (TS) framework is used to develop an algorithm that tackles WSRP. TS was chosen among other metaheuristics because it is among the most studied and mature of the metaheuristics (Glover, 1989, 1990b,a; Glover and Laguna, 1999). There is ample related literature on the success of TS when tackling combinatorial optimisation problems (Bozkaya et al., 2003; Gendreau and Potvin, 2014). In particular, TS has been widely applied to VRP and its related problems, i.e. VRPTW, MDVRP, etc., (Brandão and Mercer, 1997; Cordeau et al., 2001; Gendreau et al., 2008) which have many similarities to the WSRP. As explained in section 3.1.2 metaheuristics are high level methods which guide a series of heuristics or strategies that take advantage of the domain of the problem. As found in the previous chapter, the functions to handle time-dependent constraints were the factor that improved the results the most before applying multiple evaluations through branching. In this chapter, those functions are converted into move operators and incorporated in a TS algorithm.

In chapter 6, it was found that a local-search type procedure improves the quality of the results for the WSRP instances (GH5). It is expected that by using a well known metaheuristic such as TS, the results could be further improved. In this chapter, OpenTS (Harder et al., 2004) is used to implement the algorithm. A description of the OpenTS framework is discussed. OpenTS is developed in the Java programming language, which is the same language used to develop the elements of the Greedy

Heuristic. Given OpenTS flexibility, most of the code can be reused when developing the TS algorithm. As a result, the effort is focused on the neighbourhood moves and the parameters of the TS, rather than an implementation of the metaheuristic from scratch.

The moves used in the TS algorithm include: an insertion move for simple activities, another insertion move for activities with time-dependent constraints, i.e. complex activities; a swap move for simple activities in the same route or across different routes; a remove move that unallocates an activity and places it to the unassigned list; and a shift move that attempts to delay an activity in order to establish a new start time for the activity.

7.2 OpenTS Framework

OpenTS is a Java Framework designed to implement the TS metaheuristic. OpenTS is part of the COIN-OR (COmputational INfrastructure for Operations Research) project. The project aims to support the development of open-source software for operations research in order to speed the development and deployment of models, algorithms and computational research. In addition, the project supports the peer reviewing of its software across its users in the hope of continuously improving its tool set. Many publications have used COIN-OR software. The INFORMS (INstitute For Operations Research and Management Sciences) annual meeting has tracks that focus on contributions and applications to the COIN-OR project's tool set. For more information please refer to the project's website (www.coin-or.org)

OpenTS is flexible and easy to use as it handles all the underlying routines of a TS implementation, e.g. add/remove moves/attributes from the tabu list, acceptance criteria inclusion and event signalling at every iteration. OpenTS allows focusing on programming domain specific rules in an object-oriented manner. It requires the user to create both the solution structure for the problem to be tackled, and the neighbourhood moves, and also to define the evaluation function (Harder et al., 2004).

Figure 7.1 describes the functionality provided by OpenTS. The framework is provided with an initial solution. This solution can be obtained through an external algorithm or in many cases via a random solution generation procedure. After this initial step, the move manager interface, based on the library of moves that are provided for the specific problem, generates all possible neighbourhood moves. For example, in the case of VRPTW, a move can be a swap of two visits. As a result, the move

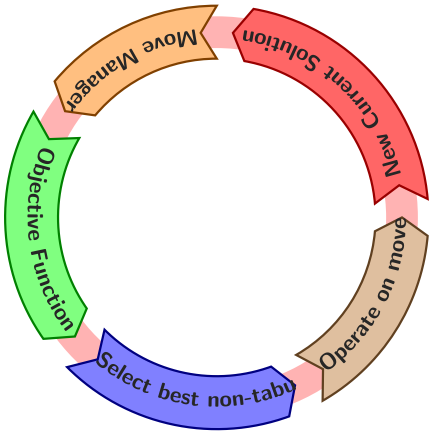

Figure 7.1: OpenTS stages during one iteration

manager creates all possible swaps between two visits according to the current state of the solution. The second stage is the evaluation of each produced move. OpenTS provides template classes (structures) to build objective function evaluators. The objective function considers the move and returns the new objective function value as if the move would have been applied. The procedure is performed for all the moves generated in the previous stage. An important factor is that at this point the structure of the solution has not changed, only the end result of the objective function is considered if a particular move is taken. The next step is to reject all moves which have been previously marked as tabu (prohibited). It could be argued that this step should be performed before evaluating the impact on the objective function in order to reduce the number of unwanted evaluations. However, the reason for discarding the tabu moves attributes after their evaluation, is the aspiration criteria. If it is the case that one move is tabu, but provides a better objective function value than the best value obtained thus far, then the move could be considered. Once the best move has been chosen, it is applied to the current solution in order to change the data structure. The evaluated objective function value becomes the current objective value and the recently changed data structure becomes the new current solution. At this point, OpenTS starts a new iteration, unless an ending criteria has been accomplished. Ending criteria are commonly based on: number of iterations, number of objective function evaluations, computation time, gap percentage and number of non-improving iterations.

7.3 Tabu Search Description

In this section the details of the implementation for the designed TS algorithm are discussed. The approach taken for the development of the TS algorithm was to follow the sample scenario provided by the documentation of OpenTS. The documentation had some helpful insights because it is based in a travelling salesman problem (TSP) (Harder, 2004).

The following are some design considerations that were followed whilst developing the TS algorithm. The considerations are important in order to understand how components of the TS algorithm function.

1. The solution structure used is the same as the one utilised in the Greedy Heuristics, i.e. a main array of employees and a list of activities which each employee performs, and an additional set node that contains the unassigned activities.

2. The procedure should start with an empty solution.

3. The objective function remains the same as the one used in chapters 5.3 and 6, that is based on penalising unassigned activities, aiming to adhere to employees' preferences and reducing the cost (travel time and distance).

4. All the constraints in the model formulation of section 5.3.2 are treated as hard constraints.

5. **Only** valid solutions are considered at all times during the searching process. A valid or feasible solution is one which satisfies all hard constraints. An empty solution, one where no activities have been assigned, is considered a valid solution.

6. As a result of the previous consideration, the designed moves guarantee producing valid neighbour solutions.

7. The number of iterations is used as one criterion for the termination of the search. However, under no circumstance any search should run for more than two hours. This two-hour parameter was established in Chapter 5 as the maximum computation time for the mathematical programming solver when tackling WSRP with daily planning horizon.

8. The moves should rely on some of the procedures developed for the Greedy Heuristic, e.g. AllocPosibleAny, Clash, Enough and ConsiderRC.

7.3.1 Neighbourhood Moves

The neighbourhood moves presented in this section expand on the work of Xu and Chiu (2001) and Mankowska et al. (2014). Xu and Chiu (2001) proposed four moves: (1) *addition* which adds an activity into an employee schedule; (2) *exchange* takes two activities, each of them assigned to different employees, and attempts to reassign them; (3) *change* reassigns a single activity to another employee; and, (4) *swap* exchanges an unassigned activity from one already assigned leaving the later unassigned. Mankowska et al. (2014) proposed four move operators for problems that contain time-dependent activities: (1) *intra-shift* and (2) *intra-swap* maintain the activities assigned to the same employee by moving one at a time or swapping two; (3) *inter-shift* and (4) *inter-swap* consider reassigning the activities to different employees. The moves distinguish two types of services i.e. activities, *single* and *double*. The latter one in the context of WSRP is an activity which needs more than one employees and/or has a time-dependency with another activity.

The moves considered in the TS must only involve feasible solutions, as a result they need to test that the process change in the solution structure maintains feasibility. Given the number of constraints in the WSRP definition, and in order to facilitate the description of the developed moves, activities are classified according to two criteria. The first criterion is whether or not the activity requires a team for its completion, i.e. more than one employee is necessary to perform the activity. The second criterion, is that an activity is *complex*, i.e. contains time-dependent constraint with another activity, or it is *simple*, if it does not have time-dependencies. There are four combinations due to the two criteria. The combinations and some considerations for each case within the moves are explained as follows:

Simple without teaming: The activities are the simplest ones regarding the design of the neighbourhood moves. These activities can be moved to any employee that has the skill to perform them and that can obey to the activities' time-window.

Simple with teaming: These activities are handled as having time-dependent constraints of the synchronisation type with their virtual copies. As a result, any neighbourhood move needs to consider that the activity requires to be assigned to more than one employee. At the beginning of the search, this multiple assignment might be easy but it becomes more difficult when all employees already have some assigned activities.

Complex without teaming: These activities have different degrees of difficulty depending on the type of time-dependent constraint they are part of. For example, a minimum time only restricts an activity to begin after some time has passed from the commencement of another one. Whereas a synchronisation type requires the same time for both activities, thus limiting the search space. Activities in this group will always require additional validations to ensure the neighbourhood moves do not violate the time-dependent constraints.

Complex with teaming: These activities are the most difficult to handle because they require consideration of other activities due to the time-dependent constraint and the additional synchronisation constraints with their own virtual activities. For example, moving an activity in this category often requires creating the space necessary in more than one employees' schedule in addition to validating that all other constraints (time windows, working time, skill-matching) remain feasible.

Every neighbourhood move can be enabled/disabled by using the appropriate parameter within the TS algorithm.

7.3.1.1 Insert

The Insert move consists in taking an activity from the unassigned list and placing it into an employee's list of activities. The move sets the activity's start time whilst ensuring that other constraints are satisfied. This move is only applicable to activities in the Simple category.

The move relies in the AllocPossibleAny procedure to find a series of candidate allocations for the activity. Each candidate allocation results in at least one possible insert move. The parameter *insert.precision* determines whether more than one insert move can be created from one candidate allocation structure, assuming the candidate allocation's flexible time is greater than zero. The parameter *insert.precision* uses the flexible time component in the candidate allocation to create different moves by shifting forward (delaying) the start time and proposing a collection of insert moves with different start times. For example, consider a call to AllocPossibleAny for an activity x that produces a candidate allocation with the following components: start time of 12:00, flexible time equal to 80 minutes, proposed assignment to Employee A, and zero idle time. If the value of *insert.precision* is 30 minutes, the produced insert moves are the following:

1. insert x in A schedule at 12:00
2. insert x in A schedule at 12:30
3. insert x in A schedule at 13:00
4. insert x in A schedule at 13:20

Four valid insert moves are proposed from the candidate allocation. Notice that the last one with start time at 13:20 does not have the same time difference of 30 minutes as the previous ones, stated in the *insert.precision* parameter, such insert move is included because 13:20 is the latest possible time allowed by the candidate allocation structure, i.e. 12:00 + 80 min = 13:20. It is generalised that regardless of the *insert.precision* parameter value, if the candidate allocation's flexible time is greater than zero, such candidate allocation will produce at least two insert moves. The first insert move has the start time indicated in the candidate allocation. The second insert move defines a new start time that is equal to the candidate allocation's start time plus flexible time. If the flexible time is zero, only one move is created.

The smaller the value of *insert.precision* the more insert moves could be generated. Careful consideration must be taken when setting this parameter to a small value. It is better to start with a big value and gradually reduce it due to the impact on the performance of the search when this move is enabled.

7.3.1.2 Insert for Time-dependent Constraints

This insert move, referred as *insertcca*, is designed to handle complex activities. It uses the AllocPossibleAny procedure to generate a list of candidate allocations. In addition, it reduces the number of candidate allocations by using the ConsiderRC procedure which validates that the time-dependent constraints are satisfied. This validation is achieved by adjusting the starting time, if necessary, and reducing the flexible time in the candidate allocation structure. If a candidate allocation cannot comply with the constraints, it is no longer considered. Once both procedures have been called, if there are still candidate allocations remaining, each of them is used to generate at least one insertcca move.

The generation of the insertcca moves from the candidate allocations uses a similar procedure as the one described for the insert move in the last section 7.3.1.1. However, the insertcca move type has its own parameter to handle the number of moves being generated: *insertcca.precision*, described in subsection 7.3.2.

The design decision to split the insert moves into two different types, one for complex and another for simple activities is based purely on performance. If a scenario does not consider time-dependent constraints, the insertcca move type can be disabled as only the insert type is necessary.

7.3.1.3 Swap

The Swap move takes two activities that are already assigned and exchanges them within the solution structure. The swap move only acts upon Simple without Teaming activities because of the number of validations that are required when exchanging two activities. As a result, if no two activities are found with such characteristic, no moves are generated. This move is inspired by other swap moves in different problems such as those for the VRP (Potvin and Rousseau, 1995; Golden et al., 2008).

The procedure to generate swap moves is as follows: firstly, all possible swaps between two activities are generated. At this point no constraints are checked. In order to generate the possible swaps moves, each assigned activity is enumerated. The maximum number of potential swaps to evaluate is given by the following expression: $maxSwaps = (assignedActivities * (assignedActivities - 1)/2$. For example, Figure 7.2 shows assigned activities that are enumerated from one to six. According to the expression, the maximum possible swap movements is: $6 * (5)/2$, i.e. 15. All resulting possible swap moves are:

1. Swap 1 for 2
2. Swap 1 for 3
3. Swap 1 for 4
4. Swap 1 for 5
5. Swap 1 for 6
6. Swap 2 for 3
7. Swap 2 for 4
8. Swap 2 for 5
9. Swap 2 for 6
10. Swap 3 for 4
11. Swap 3 for 5
12. Swap 3 for 6
13. Swap 4 for 5
14. Swap 4 for 6
15. Swap 5 for 6

Secondly, every potential swap move is tested against all relevant constraints such as skill-matching, time windows, working time, etc. If the swap move relates activities within the schedule of the same employee, e.g. 1 and 3 in Figure 7.2, the skill check is not necessary. Thus far the swap moves are incomplete because there is no indication whether both swapped activities will remain with their respective start time, or whether they can be updated accordingly. Start times are allocated to the swapped activities once it is ensured that skill-matching and that activities can be interchangeable in their respective employee's scheduled. Start times will depend on the time window configurations, swapped activities' duration and idle time availability.

Thirdly, if there could be different combinations of the start time of the swapped activities, then the parameter *swap.precision* determines the number of swap moves that can possibly be generated. For example, if swapping Act 2 and Act 5 in Figure

7.2 is valid in terms of skills, the following checks are also required:

1. Test if Act 2's duration, plus travel time from Act 4's location to Act 2's location, plus travel time from Act 2's location to Act 6's location, can fit in the time length of Act 6's start time minus Act 4's end time.

2. Test if Act 5's duration, plus travel time from Act 1's location to Act 5's location, plus travel time from Act's 5 location to Act's 3 location, can fit in the time length of Act 3's start time minus Act 1's end time.

3. If both previous tests are passed, time windows are verified for both activities.

4. Two candidate allocation-type structures are created: one for each swapped activity. The candidate allocations are used to set the start time. If flexible time is greater than zero then other combinations of start times can be generated.

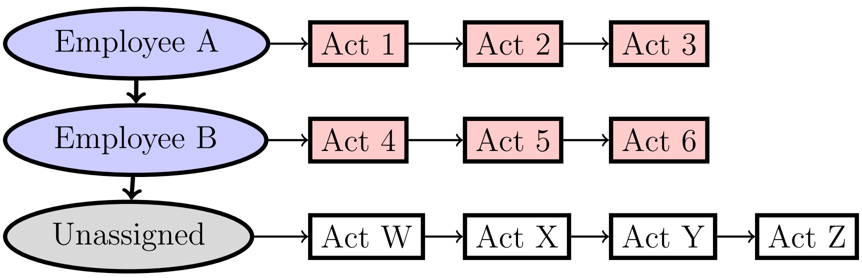

Figure 7.2: Swap move example

7.3.1.4 Remove to Unassigned

This move type removes an activity from its current employees' schedules and places the activity in the unassigned node of the solution structure. The move can be applied to all activities, i.e. complex and simple. In the case of simple without teaming activities, the procedure removes the activity and updates the employee travel and distance components as they are no longer required. It also penalises the objective function for not covering the activity within the solution. In the case of activities with teaming, the procedure needs to ensure that all employees are considered when updating travel times and distance. The procedure also ensures that only a single representation of the activity is added to the unassigned list despite the number of employees who might have had it assigned.

The remove move is used as part of a perturbation procedure when the search seems to no longer find improvements in the local neighbourhood, i.e. the algorithm is possibly stuck in some local optimum.

7.3.1.5 Shift

The shift move type delays the start time of an assigned activity. The move does not change the employee that performs the activity. It is only applicable to Simple (with or without Teaming) assigned activities. In order to generate all possible shift moves, the procedure iterates through the solution structure and tests the ability of each activity to be delayed whilst still complying to all constraints.

The procedure to generate shift moves uses the parameter *shift.precision* to determine the size of the proposed shift. For example in Figure 7.3, assume Activity 2's starting time is 12:00 pm, as shown, it can be shifted forward (orange rectangle). The shifting block length is 60 minutes. As a result, Table 7.1 lists the set of shift moves that are generated depending on the value of *shift.precision*. The values being considered for *shift.precision* are: 5 min, 10 min, 15 min and 30 min, although in the TS configuration they are inputted in milliseconds.

300000 **5 min**	**600000** **10 min**	**900000** **15 min**	**1800000** **30 min**
Shift 2 to 12:05	Shift 2 to 12:10	Shift 2 to 12:15	Shift 2 to 12:30
Shift 2 to 12:10	Shift 2 to 12:20	Shift 2 to 12:30	Shift 2 to 13:00
Shift 2 to 12:15	Shift 2 to 12:30	Shift 2 to 12:45	-
Shift 2 to 12:20	Shift 2 to 12:40	Shift 2 to 12:60	-
Shift 2 to 12:25	Shift 2 to 12:50	-	-
Shift 2 to 12:30	Shift 2 to 13:00	-	-
:	-	-	-
Shift 2 to 13:00	-	-	-

Table 7.1: Possible shift moves for Activity 2 with starting time 12:00 pm depending on the value given to the parameter *shift.precision*. Values are shown for 5 min, 10 min, 15 min and 30 min in milliseconds.

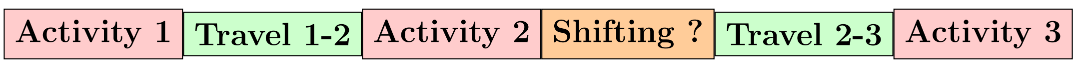

Figure 7.3: Shift move example

7.3.2 Tabu Search Parameters

The TS algorithm has other parameters apart from the ones already mentioned for the moves. The parameters can be set in order to alter the behaviour of the metaheuristic. The following list describes each parameter.

Number of Iterations (integer): This parameter establishes the number of iterations the tabu search will perform before stopping. An iteration is described earlier in the chapter in Figure 7.1. An iteration's duration varies from instance to instance, and the duration might not be constant during the search.

Time Limit (milliseconds): It restricts the amount of computation time given to the TS algorithm. The algorithm checks after the end of each iteration if it has surpassed its time limit. If it has not, then another iteration is allowed. If it has, the search process stops. The algorithm cannot guarantee that it will stop exactly at the specified time since the check is only performed at the end of an iteration which varies in duration.

Iteration Threshold (0.0 - 1.0): It is a parameter used in combination with the number of iterations. It determines the number of non-improving iterations that can occur before partially restarting the allocation process. For example, if the number of iterations is 1000, and the iteration threshold parameter is set to 0.2, then $1000 \times 0.2 = 200$. Non-improving iterations are allowed to occur before a diversification method is used. When the iteration threshold is reached it is assumed that the search has been trapped in local optimum. Once the diversification method is triggered the internal counter is restarted. Every time a new best solution is found the counter is reset as well.

Forced Remove (0.0 - 1.0): It is the probability that each assigned activity has to be removed and placed in the unassigned list during the diversification method. The diversification method consists in iterating through all assigned activities and testing given this probability if each activity is unassigned.

The tabu list is implemented as multiple lists, one for each of the move operators (see Section 7.3.2.3). Therefore, there is an initialisation parameter for the size of each list:

Initial Insert Tenure (integer): sets the initial tabu list size for the insert-type moves. It includes insert and insertcca.

Initial Remove Tenure sets the initial tabu list size for the remove type moves.

Initial Swap Tenure (integer): sets the initial tabu list size for the swap type moves.

Initial Shift Tenure (integer): sets the initial tabu list size for shift type moves.

The next set of parameters enabled/disabled the use of a specific move.

insert.include (boolean): It determines if the insert move is enabled.

insertcca.include (boolean): It determines if the insertcca move is enabled.

remove.include (boolean): It determines if the remove move is enabled.

swap.include (boolean): It determines if the swap move is enabled.

shift.include (boolean): It determines if the shift move is enabled.

Update Tenure After (integer): This parameter represents the number of iterations that must pass before the tenure of all tabu lists is updated. After this number of iterations have passed, each tabu list tenure is updated to the average number of movements generated for each type during the x most recent iterations. Where x is the initial tenure of each tabu list.

insert.precision (milliseconds): It determines the number of insert moves depending on the presence of flexible time in the candidate allocation structure used to generate the move.

insertcca.precision (milliseconds): It determines the number of insertcca moves depending on the presence of flexible time in the candidate allocation structure used to generate the move.

swap.precision (milliseconds): It determines the number of swap moves depending on the presence of flexible time in the candidate allocation structure used to generate the move.

shift.precision (milliseconds): It determines the number of shift moves depending on the presence of flexible time in the candidate allocation structure used to generate the move.

7.3.2.1 Evaluation Function

The evaluation function is the same utilised in the MIP and Greedy Heuristic (see Equation 5.12). The values of the weights ω_1, ω_2, ω_3 are calculated as described in Equations 5.25, 5.26 5.27. The weights change the emphasis on a given component within the evaluation function, which comprises of three components: cost, employees' preferences and assigned activities. Therefore the weights are calculated per instance. Only ω_1 remains the same for all instances with a value of (1). The weight ω_2 is the sum of all assignations and weight $\omega 3$ is ω_2 times the number of visits times the value of the maximum preference within the instance. The values can be computed in advance as the data required is known before the searching process commences. Due to the number of evaluations of this function, it was necessary to implement delta functions, i.e. increments/decrements values that could be added/subtracted from the objective function to update its value. The use of delta functions avoids having to compute the objective value from scratch.

7.3.2.2 Initial solution

The algorithm can start from an empty solution, i.e. no activities assigned, as it is still considered a valid solution. Alternatively, the algorithm can start with a solution provided by another heuristic. However, the algorithm cannot start with a randomly generated solution, unless the solution is tested for feasibility first. A random assignment of activities is unlikely to generate a feasible initial solution. As a result, random initialisation could compromise one of the design considerations of this TS which is to always maintain feasibility of solutions throughout the searching process.

If the TS starts with an empty solution it uses the insert and insertcca moves until no further inserts can be made. A successful insert move reduces greatly the objective function as the penalty for unassigned activities gets smaller. It should be remembered that the weight associated with unassigned activities is the biggest one (see section 5.3.3.1).

In early experiments the TS was obtaining better results if started with an empty solution than when initialised with the best solution obtained by the Greedy Heuristic. This observation motivated the addition of a diversification (partially destroying) the current solution after a number of iteration without improvement.

7.3.2.3 Tabu tenure

The tabu tenure refers to the size of the tabu list. In OpenTS a basic interface to create a tailored tabu list is provided. The implementation of the list is based on storing the moves that have been used recently. Moves' storage is performed by keeping the hash code of the move in an array. The verification of whether a move is tabu or not is performed by comparing the hash code of the candidate move to the stored hash codes. Hash code encoding has the advantage of being quick to verify as only comparison between two integers values is required. The disadvantage is the loss of information when performing the encoding which could lead to two moves having the same hash code. Additionally, in some cases, two moves that have the same outcome, have different hash codes, thus potentially allowing a move that should have been banned. For example, Swap 1 for 2 and Swap 2 for 1 have different hash codes but clearly the moves are equivalent.

The tabu list is implemented as multiple lists. One list for each type of neighbourhood move. However, insert and insertcca share the same tabu list. A multiple list approach allows better control on which moves can be enabled/disabled during the search. In a single list approach, two different types of moves can have the same hash code, thus by each type of move having its own list such a scenario is prevented. Multiple lists also allow for different sizes of lists for each type of move. For example, at any iteration, the maximum number of Remove moves marked as tabu should never be greater than the number of activities in the instance. Whereas the number of shift type moves, when using a small $shift.precision$ value, could be thousands. Another advantage of multiple lists is when disabling a move during the search process, the tabu list of the disabled move can be "frozen", i.e. its state conserved, for when it is enabled again. Otherwise, if using a single tabu list, moves from the disabled type remain in the list until enough iterations have passed to drop them out entirely.

All tabu lists are dynamically adapted after a number of iterations defined by **Update Tenure After** parameter. The size of each tabu list is determined by the average number of moves that were generated for evaluation in the last X iterations (X = **Update Tenure After** parameter value). For example, if the procedure to generate all possible shift type moves has had the following historic number of different shift moves previously generated: 278, 375, 267, 300, 434 and the Update Tenure After parameter is set to five, then after five iterations the tenure of the shift tabu list is $(278+375+267+300+434)/5$, i.e. 330. The purpose of such modification in the size of the tabu lists is to help the search process. It is expected that if recurrent iterations have generated larger sizes of different moves then the corresponding move's tabu list remains large allowing the use of as many as possible. On the opposite, if the number of moves is small then the size of the tabu list reduces allowing moves to be reused quicker. The procedure is inspired from the adaptive tabu tenure of Devarenne et al. (2008).

7.3.2.4 Aspiration Criteria

The aspiration criteria used in the TS algorithm is Best Solution Found. In other words, if applying a move that is currently marked as tabu generates a new best objective value, such move is allowed despite being prohibited.

7.3.2.5 Perturbation

The TS incorporates a perturbation function that allows the search to partially restart after a period in which no improvements are made. In other TS implementations the diversification is obtained by handling the tabu tenure, aspiration criteria and tabu restrictions. In such applications infeasible solutions tend to be allowed which help the search to escape local optima. Given the design decision of not allowing infeasible solutions at any stage during the search, there are cases in which the algorithm gets stuck in local optimum despite: using multiple tabu lists and dynamic resetting of the tenures. After noticing such behaviour, the introduction of a perturbation stage that allowed the search to partially restart was introduced. As described earlier, the perturbation consists of testing whether each assigned activity is to be removed subject to the probability given by $ForcedRemove$ parameter.

7.3.2.6 Stop criteria: # of Iterations and Computation Time

The number of iterations and computation time are the two parameters the TS uses to stop exploring for better solutions. In the experiment settings different values for number of iterations are explored. In respect of computation time there is a maximum limit established of 2 hours. The limit was decided in order to match the maximum computation time allowed to the solver. Therefore, there are two termination criteria for the tabu search: the tested number of iterations or two hours of computation time, whichever occurs first.

7.4 Experimental Results

The objective of this set of experiments is to detect the best parameter settings for the TS algorithm yielding the best results considering all problem instances. Table 7.2 contains the parameter value settings being considered. The value for each parameter is fixed incrementally investigating how one parameter setting is affected by the setting of others. There are other parameters within the TS algorithm that remain the same. For example, all moves were allowed at all times thus insert.include, insertcca.include, remove.include, swap.include and shift.include are enabled. All tabu tenures are initialised with a value of 10 (Initial Insert Tenure, Initial Remove Tenure, Initial Swap Tenure and Initial Shift Tenure).

Nine configurations of parameters were chosen using the possible values of Table

7.2. The nine configurations were not set in advance. It was an exploratory set of parameters. In other words, given the results of the initial configuration the next configuration was decided based on some analysis which is described in the following paragraphs. The results of each configuration are compared to the best known results for each instance. The best known results are the ones obtained either though the mathematical solver or any version of the Greedy Heuristic.

Parameter	**Values**
Number of Iterations	1000, 10000, 50000, 100000
Time Limit	1 hour, 2 hours
Iteration Threshold	0.0001, 0.025, 0.05, 0.1, 0.2
Forced Remove	0.3, 0.5, 0.75, 1.0
Update Tenure After	10, 20, 50, 100
insert.precision	1 minute, 10 minutes, 15 minutes, 30 minutes
insertcca.precision	1 minute, 5 minutes, 10 minutes, 15 minutes
swap.precision	1 minute, 5 minutes, 10 minutes, 15 minutes
shift.precision	1 minute, 5 minutes, 10 minutes, 15 minutes

Table 7.2: Parameter value settings.

The first configuration of values (see Table 7.3) was chosen based on experience of the performance of the greedy heuristic and common values encountered in the TS literature of VRPTW. For example, a common value for the number of iterations is 10000 as in (Cordeau et al., 2001). The time limit was set equivalent to experiments with the mathematical solver (2 hours). The iteration threshold at 5% meant that if after 500 iterations the best result does not improve, the perturbation method is initiated. A value of 50% probability of being removed for each assigned task guarantees enough perturbation without destroying the solution. Setting the tenure update after 100 iterations limits the maximum number of tenure adjustments to 100. The value for the precision of insert moves (30 min) allows few evaluation moves at the beginning of the search as in most of the instances the planning horizon is one day. In the worst case if the time window of an activity matches the planning horizon, it means 48 possible moves for such activity. For insertcca the precision was five minutes because the time-dependent constraints already restrict greatly the possible time assignment. Thus, opting to increase the number of this type of moves. Swap and shift precision were set arbitrarily.

The results for each instance using the initial configuration can be found in the Appendix E.1. Overall, when compared to the best known results of an instance, the TS algorithm finds improvements for 82 instances out of the 375. The initial configuration obtains feasible solutions for only 373 instances. Two ran out of memory without even creating a first iteration due to the number of moves that are required to be analysed. The objective function value of these two instances is reported as

Parameter	Value	Parameter	Value
Number of Iterations	10000	insert.precision	30 minutes
Time Limit	2 hours	insertcca.precision	5 minutes
Iteration Threshold	0.05	swap.precision	10 minutes
Forced Remove	0.50	shift.precision	5 minutes
Update Tenure After	100		

Table 7.3: Parameter values for the first configuration.

empty solution which is still a valid one. 352 instances finished all iterations (10000) but 16 timed out after achieving two hours of search. The average computation time for those instances which completed all the iterations was 1022 seconds.

Parameter	Value	Parameter	Value
Number of Iterations	1000	insert.precision	30 minutes
Time Limit	2 hours	insertcca.precision	5 minutes
Iteration Threshold	0.05	swap.precision	10 minutes
Forced Remove	0.50	shift.precision	5 minutes
Update Tenure After	100		

Table 7.4: Parameter values for the second configuration.

A second configuration was used (see Table 7.4). The individual results for all instances obtained using the second configuration can be found in the Appendix E.2. This configuration used the majority of the same values as the initial configuration but reduced the number of iterations to 1000. The reason was to verify the quality of solutions obtained by decreasing the iterations. It could be argued that observing the trace of the objective function over time from the results of the initial configuration could provide such verification. Nevertheless, the number of iterations before perturbation (obtained through the Iteration Threshold parameter) depends on the number of iterations defined, and by reducing the number of iterations from 10000 to 1000, we also affect the number of iterations before perturbation to 50. Therefore, a simple verification only by tracing might not yield the same results. Overall, only 32 instances obtained better results when compared to the best known solutions. The maximum time used for an instance to complete the 1000 iterations was 6053 seconds whereas the minimum was 2 seconds.

It was observed that decreasing the number of iterations reduces the number of best solutions from 82 to 32. Therefore it is reasonable to conclude that a third configuration with the same parameter values but increasing the number of iterations to 100000 (tenfold the initial configuration) might also increase the number of best solutions. The third configuration (see Table 7.5) obtained 100 best solutions. Going from 1000 to 10000 iterations increase the number of best solutions more than double (32 to

Parameter	Value	Parameter	Value
Number of Iterations	100000	insert.precision	30 minutes
Time Limit	2 hours	insertcca.precision	5 minutes
Iteration Threshold	0.05	swap.precision	10 minutes
Forced Remove	0.50	shift.precision	5 minutes
Update Tenure After	100		

Table 7.5: Parameter values for the third configuration.

82). But applying the same rate of increment again only yield an increase of 10% (82 to 100). Moreover, it was observed that the number of instances reaching the time limit before completing the 100000 iterations was 165 which is ten times more than in the first configuration. It could be argued that if more instances had finished all their iterations, perhaps there might be more than 100 best solutions. Nonetheless increasing the time limit is not an option as the maximum value for computational time established was two hours. As a result the remaining of the configurations are aimed at testing other parameters that have remain the same.

The minimum computation time for an instance, using the third configuration, that completed all iterations was 35 seconds, a maximum of 7122 seconds (almost two hours) and average of 2054 seconds (34 min). The individual results for each instance can be found in the Appendix E.3.

Parameter	Value	Parameter	Value
Number of Iterations	1000	insert.precision	15 minutes
Time Limit	1 hour	insertcca.precision	1 minutes
Iteration Threshold	0.10	swap.precision	5 minutes
Forced Remove	0.75	shift.precision	5 minutes
Update Tenure After	10		

Table 7.6: Parameter values for the fourth configuration.

The fourth configuration (see Table 7.6) considers 1000 number of iterations. It reduces the computation time to one hour (3600 seconds) and increases the threshold to 10% rather than 5% as in the last three configurations. This setting gives more iterations to explore local areas before perturbation might occur. In the worst case, this configuration creates ten perturbations. Since the number of perturbations is reduced, the probability of an activity being unassigned increases to 0.75, as a way of compensating for the decrease in perturbation cycles. The idea is to make sure that when a perturbation occurs, the change in the solution structure is more drastic than in previous configurations, thus increasing the chance of moving away to a different area of the search space. In addition, the number of times the tabu tenures are adjusted is increased by reducing the update tenure parameter to ten. Some precision parameters are changed: insert.precision is set to 15 min, insertcca.precision set to 1 min and swap.precision set to 5 min. Those settings have the effect of producing more moves in every iteration. When comparing the fourth configuration to the second one 14 instances ran out of memory during the search after producing some feasible results in comparison to two instances. 343 instances finish all 1000 iterations before the hour of computation time and 12 instances ran of time before completing 1000 iterations. In the second there was no instance reaching the time limit before completing the iterations which confirms that reducing the precision parameters creates more moves to consider and increases the time spent in completing one iteration. All the results of using the fourth configuration in the TS across all instances are in the Appendix E.4

Parameter	**Value**	**Parameter**	**Value**
Number of Iterations	1000	insert.precision	15 minutes
Time Limit	1 hour	insertcca.precision	5 minutes
Iteration Threshold	0.025	swap.precision	5 minutes
Forced Remove	0.750	shift.precision	1 minutes
Update Tenure After	50		

Table 7.7: Parameter values for the fifth configuration.

The fifth configuration (for individual results refer to Appendix E.5) increases the insertcca.precision parameter back to 5 min and changes the shift.precision to 1 minute whilst maintaining the same time limit and number of iterations. It sets the update tenure parameter to 50, but reduces the percentage of iterations that must be passed with non-improving results to trigger the perturbation to a value of 0.025. Such configuration reduces the instances that ran out of memory to three. 353 instances finish 1000 iterations within one hour and 15 instances ran out of time before achieving 1000 iterations. Nevertheless, this configuration only achieves 5 best solutions when compared to the best known, the lowest number among all the configurations considered until now.

Parameter	**Value**	**Parameter**	**Value**
Number of Iterations	1000	insert.precision	1 minute
Time Limit	1 hour	insertcca.precision	1 minute
Iteration Threshold	0.20	swap.precision	1 minute
Forced Remove	0.75	shift.precision	1 minute
Update Tenure After	20		

Table 7.8: Parameter values for the sixth configuration.

It was observed that decreasing the value of the precision parameters increases the number of moves. In some cases the moves are so many that the program runs out of memory. To verify this observation a sixth configuration sets the precision parameters to their lowest value, 1 minute, for all four precision parameters. The rest of the parameters are: number of iterations 1000; computation time one hour; iteration threshold 0.20; forced remove 0.75; and number of iterations for tenure adjustment 20. This configuration only produces results for 271 instances, the rest (103) ran out of memory without producing any preliminary feasible results. 58 out of the 271 ran out of memory after producing a preliminary result. Therefore, increasing the number of moves in one iteration might lead the algorithm to run out of memory. The memory issue could be solved by increasing it or with a different implementation of the TS which handles memory more efficiently. Only 169 instances complete the 1000 iterations within one hour. 44 instances ran out of time before completing the 1000 iterations. Results are available in Appendix E.6

Parameter	**Value**	**Parameter**	**Value**
Number of Iterations	1000	insert.precision	10 minutes
Time Limit	1 hour	insertcca.precision	10 minutes
Iteration Threshold	0.01	swap.precision	10 minutes
Forced Remove	0.90	shift.precision	10 minutes
Update Tenure After	25		

Table 7.9: Parameter values for the seventh configuration.

A seventh configuration of parameters is considered. In this occasion all precision parameters are set to ten minutes. The number of iterations is 1000 and one hour is assigned as time limit. The iteration threshold is reduced to 0.01, thus allowing for more perturbations cycles as only 10 non-improving iterations must pass. Also, the probability of removal is set to 0.90 and the adjustment of tenures is configured after 25 iterations. The seventh configuration aims to scan different regions of the search space as not enough time is spent in any region and the perturbation is so significant that it is almost like a full restart. 368 instances obtain results. The rest ran out of memory without finishing a single iteration. Among the 368, no instance runs out of memory, instances either complete the 1000 iterations within time (334) or finish the computation time (34). The result of this configuration for all instances can be found in the Appendix E.7. This configuration yields the worst results so far in terms of best solutions found, only two.

The eighth configuration seeks to evaluate the lowest iteration threshold assigned (0.0001). The number of iterations is defined as 50000 which allows only five non-improving iterations before a perturbation cycle starts. However, the probability of an activity to become unassigned is set to 0.30 (forced remove) in order to maintain the majority of the solution's structure. Tenures are updated after 50 iterations. Precision parameters are maintained at ten minutes, with the exception of *swap.precision*,

Parameter	**Value**	**Parameter**	**Value**
Number of Iterations	50000	insert.precision	10 minutes
Time Limit	1 hour	insertcca.precision	10 minutes
Iteration Threshold	0.0001	swap.precision	15 minutes
Forced Remove	0.3000	shift.precision	10 minutes
Update Tenure After	50		

Table 7.10: Parameter values for the eighth configuration.

which value is 15 minutes. 368 instances obtain feasible results. The rest ran out of memory without finishing one iteration. Out of the 368, no instance ran out of memory. The instances either complete the 50000 iterations (161) or ran out of computation time (207). Individual results per instances are available in the Appendix E.8.

Parameter	**Value**	**Parameter**	**Value**
Number of Iterations	10000	insert.precision	15 minutes
Time Limit	1 hour	insertcca.precision	15 minutes
Iteration Threshold	0.025	swap.precision	15 minutes
Forced Remove	1.000	shift.precision	15 minutes
Update Tenure After	10		

Table 7.11: Parameter values for the final configuration.

The final configuration sets the number of iterations to 10000, computation time to one hour. A value of 0.025 is assigned to the iteration threshold. Such value allows 250 non-improving iterations before a perturbation cycled can be called. The probability for being removed is set to 1.0, in other words, producing a complete restart. Tabu tenures are adjusted after ten iterations. The precision parameters are all set to 15 minutes. This configuration obtains 371 instances with feasible results, only 3 ran out of memory without performing an iteration. One instance ran out of memory after some iterations. 320 instances finish the 10000 iterations and 50 ran out of computation time. Refer to Appendix E.9 for each instance' results using the ninth configuration of parameters for the TS.

Table 7.12 shows a summary of the experiments' results. It divides the results of each configuration into four groups: 1) Out of Memory with no iterations performed (empty solution); 2) Out of Memory with intermediate valid solutions; 3) Instances where all iterations were completed, in accordance with its parameter value configuration; and 4) Instances which reach the time limit but do not complete all iterations but have some intermediate results. When appropriate each group presents descriptive statistics (minimum, maximum, mean and standard deviation) on the number of iterations and computation time. Focusing on the first group the relevant fact is that in the sixth configuration 103 instances did not complete a single iteration due to the number of moves generated. In the second group, two configurations (7th and 8th) did not experience problems with memory once at least one iterations had passed. Even though the number of iterations is different, one having 1000 and another 50000 the common parameters values are those involved with the number amount of moves generated per iteration. In the third group the fourth configuration seems to have a good balance of best solution achieved (21), time limit of one hour with almost all instances (343) finishing the 1000 iterations using an average time of 8 minutes. In the fourth group the second configuration seems to be the best as the set of parameters allow all instances that did not ran out of memory to finish within the time limit. But, once againg like in the third group, the fourth configuration is offers good results as well using half of the time.

Result	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
No iterations(OoM)	1	1	1	5	3	103	6	6	3
Success	373	373	373	369	371	271	368	368	371
OoM with iterations	5	2	1	14	3	58	0	0	1
Min time (seconds)	543	254	3456	225	277	7	-	-	1244
Max time (seconds)	2030	320	3456	2131	2727	3401	-	-	1244
Mean time (seconds)	1229.00	287.00	3456.00	760.00	1282.66	709.32	-	-	1244.00
StDev time (seconds)	621.58	33.00	0.00	528.62	1047.20	882.62	-	-	0.00
Min iterations	718	248	5218	21	21	5	-	-	4002
Max iterations	5089	811	5218	981	369	884	-	-	4002
Mean iterations	3080.00	529.50	5218.00	589.78	163.66	119.53	-	-	4002.00
StDev iterations	1398.00	281.50	0.00	291.93	148.82	199.58	-	-	0.00
All iterations	352	371	207	343	353	169	334	161	320
Min time (seconds)	5	2	35	2	3	4	3	24	6
Max time (seconds)	6359	6053	7122	3104	3478	3284	3583	3590	3525
Mean time (seconds)	1022.52	324.00	2054.03	487.15	600.97	839.10	518.78	1330.80	804.07
StDev time (seconds)	1243.68	618.68	2004.49	657.26	688.91	884.71	877.98	1048.43	1002.64
Run out of Time	16	0	165	12	15	44	34	207	50
Min iterations	2123	-	3657	243	213	135	392	518	1745
Max iterations	8865	-	99572	913	983	975	990	48982	9880
Mean iterations	5118.75	-	43980.63	549.91	639.73	544.79	716.35	14753.90	6620.96
StDev iterations	2194.85	-	26182.28	202.25	257.49	253.53	168.85	12723.20	2553.09

Table 7.12: Descriptive statistics for computation time and number of iterations for each of the nine configurations of parameters tested. The first section provides overall number of instances for which the configurations achieved feasible results. The second section groups the instances that ran out of memory but had already provided feasible solutions, time is also provided. The third section shows descriptive statistics on the instances that completed the iterations given in their configurations. The fourth section groups the instances that did not complete the iterations and utilised all the time limit specified.

Tables 7.13 and 7.14 show the minimum, maximum and mean gap for each of the nine configuration of parameters. The gap is calculated against the best known result so far for each instance, i.e. either through the mathematical solver or any version of the Greedy Heuristic. Gap excludes instances that ran out of memory without a single iteration. The results are divided into two groups for each of the nine parameter configurations. The first group presents gap statistics on those instances for which the TS obtained better results, i.e. a reduction in the objective function value. The second group presents similar descriptive statics for the set of instances where the previously known results are better. In the first group the gap is from the previous results to the new ones obtained by the TS. In the second group the gap is from the TS's results to the best known.

The ninth configuration obtain the best average gap (146%) when compared to the best known results. For all configurations the secong group shows that at least one instance obtain almost the same results as the best known.

Concept	**1**	**2**	**3**	**4**	**5**
Total	373	373	373	369	371
Tabu Search	82	32	100	21	7
MinGap	0.1419%	0.1064%	0.4838%	0.4998%	0.0003%
MaxGap	290.8042%	153.1308%	329.7287%	153.1365%	7.1658%
MeanGap	35.2706%	13.8169%	36.0135%	14.0964%	2.2323%
StdGap	69.1821%	27.5752%	75.5370%	31.7691%	2.2757%
Solver OR GH	291	341	273	348	364
MinGap	0.0000%	0.0000%	0.0000%	0.0000%	0.0000%
MaxGap	15435.4673%	12188.5217%	15285.6225%	12138.5745%	15335.5726%
MeanGap	147.6775%	171.8170%	166.1693%	180.7306%	249.4354%
StdGap	971.0024%	854.6415%	1037.9040%	866.2459%	1179.3180%

Table 7.13: Gap results for all tabu search experiments

Concept	**6**	**7**	**8**	**9**
Total	271	368	368	371
Tabu Search	38	2	22	29
MinGap	0.0177%	0.7781%	0.2386%	1.0215%
MaxGap	147.5379%	2.5250%	17.1611%	186.5938%
MeanGap	24.8566%	1.6515%	7.1545%	24.4333%
StdGap	34.0050%	0.8734%	6.3988%	42.9280%
Solver OR GH	233	366	346	342
MinGap	0.0000%	0.0000%	0.0000%	0.0000%
MaxGap	16084.8698%	19531.6276%	12740.5511%	15335.5754%
MeanGap	964.8761%	275.5043%	222.2268%	146.3689%
StdGap	2245.2111%	1333.6382%	990.4192%	930.5249%

Table 7.14: Gap results for all tabu search experiments

7.5 Conclusion

Table 7.15 contains the parameters setting of each of the nine configurations.

Parameter	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
number of iterations	10000	1000	100000	1000	1000	1000	1000	50000	10000
computation time	7200	7200	7200	3600	3600	3600	3600	3600	3600
iteration threshold	0.0500	0.0500	0.0500	0.1000	0.0250	0.2000	0.0100	0.0001	0.0250
forced remove	0.5000	0.5000	0.5000	0.7500	0.7500	0.7500	0.9000	0.3000	1.0000
update tenure after	100	100	100	10	50	20	25	50	10
insert.precision	30 min	30 min	30 min	15 min	15 min	1 min	10 min	10 min	15 min
insertcca.precision	5 min	5 min	5 min	1 min	5 min	1 min	10 min	10 min	15 min
swap.precision	10 min	10 min	10 min	5 min	5 min	1 min	10 min	15 min	15 min
shift.precision	5 min	5 min	5 min	5 min	1 min	1 min	10 min	10 min	15 min

Table 7.15: Setting of parameters for each of the nine considered configurations for the TS.

Using 1000 iterations obtains best results for 38 instances (Config. 6), but also as few as 2 instances (Config. 7). The difference between those two configurations is the precision parameters. Config. 6 uses lower values for the precision parameters, which increases the diversity of moves. It allows the moves to explore time differences of one minute. Whereas in Config. 7, the moves are restricted to ten-minute variations. Config. 8 uses similar precision parameters but increases iterations to 50000. The increase on iterations obtains 22 instances with better results, but is still lower than 38 obtained by only using 1000 iterations with small precision values (Config. 6). It is clear that more iterations can help to obtain better results as shown in Configurations 1, 2, and 3 where the parameters remain the same, only adjusting the number of iterations tenfold. Findings for Config. 2 (1000 iterations) is 32 best results; for Config. 1 (10000 iterations) 82 best results; and for Config. 3 (100000 iterations) 100 better results. However, when using 100000 iterations, 45% percent of the instances do not finish within the two hour limit. Two hours is the computation time that guarantees all instances completing at least 1000 iterations. In configurations where computation time was limited to one hour and as low as 1000 iterations, there were still instances that ran out of computation time (Configurations 4, 5, 6, and 7).

Configurations 4, 5, 6 and 7 all have 1000 iterations and one hour computation limit. Out of those configurations, the worst obtaining best results is Config. 7 with only two and the best is Config. 6 with 38. Configurations 4 and 5 can be considered transitional configurations between Config. 7 towards Config. 6. Configuration 4 reduces (insertcca.precision) to one minute and obtains 21 instances with best results. Whereas Configuration 5 reduces (shift.precision) and obtains only 7 instances with best results. It seems that insertcca.precision, if set as low as possible, produces better results. This could be because the move type insertcca is the only move that acts on activities with time-dependent constraints. As a result, it helps to find better arrangements in the solution structure for such activities with time-dependent constraints. It was experimentally proven that such constraints make the WSRP instances more difficult to tackle (see 5.2.3.3). Thus, the behaviour of the TS, judging from the results obtained, seems to indicate that the move type insertcca which focuses on time-dependent activities constraints, is more useful when finding best results.

The overall conclusion in terms of the TS implementation is that increases to the number of iterations increases the quality of the results obtained. However, after 10000 iterations some instances take a great deal of time to improve the quality of the results. It seems that the parameter *insertcca.precision*, which controls the number of insert moves for activities with time-dependent constraints, makes a difference on producing better results when set with low values.

In terms of the continuation of the work of previous chapters, it can be concluded that the tailored functions, developed to tackle activities with time-dependent constraints for the greedy heuristic (versions GH3 - GH5), work well when transformed into neighbourhood moves. The results obtained by using such neighbourhood moves in a Tabu Search implementation confirm it.

Chapter 8

Conclusions and Future Work

8.1 Introduction

This thesis presents optimisation models and algorithms to tackle Workforce Scheduling and Routing Problems (WSRPs). The WSRP considers a set of employees who are required to travel across multiple customer locations in order to perform job related activities. Each employee can have different skills and qualifications which determine the activities that the employee can perform. In addition, employees can have different starting and ending locations for a working day. For example, employees could start from the same location, i.e. the organisation's main office, and end their working day by returning home, or alternatively they could start their working day from home. In WSRP, employees are not subject to the same means of transportation. The most common modes of transportation are: private vehicles, company vehicles, public transport, e.g. bus or train, bicycle, and by foot, i.e. walking. The activities vary in terms of skills requirements. As a result, a skill-matching between the activities and employees is needed. Activities have associated time windows which dictate the activities' possible starting time. Time windows should be respected when assigning employees. In addition, some activities might require more than one employee, i.e. a team. Finally, activities might also have time-dependency relationships with other activities. The time-dependencies can be of five different types: synchronisation, overlap, minimum difference, maximum difference and minimum-maximum (min-max) difference. The WSRP combines features from the general employee scheduling problem and from vehicle routing problems. This combination of scheduling and routing makes it a hard combinatorial optimisation problem.

8.2 Review of Contributions

In this section a revision of the main contribution of this thesis is provided.

8.2.1 WSRP Data Sets

Five data sets were obtained from different WSRP-like problems. The data sets were adapted to reflect the main characteristics of a WSRP. The adaptation generated 375 instances. The data sets were presented in Chapter 4. An analysis of the configuration of the data sets was performed. The analysis confirmed the diversity regarding instances with different number of employees, activities and configurations of time windows.

The complete data set is available at: `http://www.cs.nott.ac.uk/~jac/dataset.html`

8.2.2 Mathematical Models

Two mathematical models were adapted from the literature to address WSRP. The first model, an Integer Linear Programming (ILP), focused on assigning all activities listed in an instance. The ILP objective function included the cost of assigning employees to activities and the travel time. A mathematical solver, Gurobi, was used to solve the model in a subset of the instances. For 50% of the instances the solver could not provide feasible results. The reason was because some instances were understaffed as there were not enough employee-working hours to cover all activities. In addition, for the instances where the solver could find optimal solutions, the majority of the gap reduction (90%) was performed during the first two hours of computation time. The figure of two hours was adopted as the maximum computation time allowed because it is a reasonable time to wait for a solution in a daily problem. The second model, a Mixed Integer Linear Programming (MILP), allows activities to be left unassigned by incorporating a penalty cost in the objective function. This change meant that all instances could be used including the understaffed ones. The MILP also incorporated employees' preferences on activities. The MILP objective function included a cost of assignment defined by travel time and distance, employees' preferences and the penalty for unassigned activities. Activities were given a priority level in order to address activities that favour the assignment of emergency activities over low priority ones. A benchmark of results was produced that included feasible solutions for 338 instances. The solver runs out of memory in the remaining instances due to their size in terms of number of activities and employees.

8.2.3 Teaming Representation

Synchronisation constraints were used in activities that require a team of employees forcing them to arrive at the same time. The procedure consists of creating virtual copies of an activity with a team requirement. The virtual activities have the same requirements as the original one. Then a synchronisation constraint is enforced for every resulting pair between the original and its virtual activities. Such an approach increased the size of the model as it incorporates more activities resulting in an expanded network. A reduction in the number of variables in the mathematical model was introduced. The reduction eliminated edges in the underlying network. The removed edges represent unrealistic transitions between an activity and its virtual counterparts. Employees performing the original activities cannot transit to the virtual ones as they represent the same thing. As a result, the variables in the model that represent such edges can be eliminated for all employees, reducing the model size.

8.2.4 Greedy Heuristic for WSRP

The most difficult of the contributions was the designed and development of a greedy heuristic (GH) for the WSRP. GH was designed to use as much as possible the information provided by the instances to quickly identify configurations that lead to good feasible results. Five versions of GH were discussed. GH1 was inspired by the bin-packing problem. GH2 expands the searching space available to assign activities by including intermediate idle times. GH3 incorporates tailored functions for each of the five types of time-dependent constraints which lead to obtaining feasible solutions for all instances. GH4 introduces a catalogue of allocation options when assigning activities. Finally GH5 used branching in order to copy the solution structure and investigate more than one allocation option. The different versions of GH rely on tackling time-dependent activities as soon as they appear, in other words, prioritising complex activities. The tailored functions for each type of time-dependent constraints were difficult to design as they had to consider all other constraints when evaluating allocation options.

8.2.5 Tabu Search Implementation

A Tabu Search (TS) implementation using OpenTS was developed to tackle WSRP. OpenTS is a Java framework that supports the development of TS algorithms. OpenTS provides the interface to handle the manipulation of the tabu list, the evaluations of the objective function and other events occurring during an iteration of the algorithm. The implementation efforts focused on five neighbourhood moves: insert, insertcca, remove, swap and shift. The moves maintain feasible solutions when applied to a known solution, if possible. The moves could also be used to construct a feasible solution starting with no activities assigned. The TS implemented a multiple tabu list approach. For each type of move one tabu list was used. Such an approach allowed more flexibility when enabling/disabling the usage of move types. A mechanism was introduced that adjusted the tabu tenures of the lists based on the mean number of feasible moves being created in previous iterations. The number of previous iterations can be changed as it is a parameter that needs to be defined. Every neighbourhood move has an associated precision parameter that decreases/increases the number of potential moves that are generated during an iteration of the TS. Those precision parameters can be used to intensify the search. Despite these parameters the TS was getting trapped in local optimum. To avoid this, a diversification mechanism was introduced that partially destroyed the solution by unassigning activities subject to a probability. The diversification mechanism is triggered after a number of non-improving iterations has completed. Both the probability of unassignment and the number of non-improving iterations are parameters that can be set in the TS. Nine different configurations of parameters were tested. The experiments show that the more iterations performed the better results can be found, but after 10000 iterations the improvements take much longer to occur. The duration of an iteration is not constant as it depends on the number of available moves that require evaluation in an iteration. The number of moves also relates to the size of the instance and the parameters being used. In such circumstances it was shown that only 1000 iterations for all instances could be guaranteed in two hours of computation time. The TS runs out of memory in some instances because of the number of moves that are generated in an iteration, a characteristic also present in the experiments with the mathematical solver. It was found that the insertcca move type yields better results when configured to small values. Such move type is the only one dealing with activities with time-dependent constraints, which reinforces previous findings that tackling this kind of activities first leads to significant improvements in the overall results.

8.2.6 Benchmark Results

The best results obtained by the mathematical solver (MILP model), the greedy heuristic and the tabu search implementation provide a benchmark that will facilitate future comparisons for other solution methods for the WSRP. Three publications have already used the benchmarked results to some extent.

8.2.6.1 Best Results

Chapters 5.3, 6, 7 focused on different methods of tackling WSRP. The methods included mathematical programming, a greedy heuristic and tabu search. Such chapters used the same objective function and instances. Table 8.1 presents the number of instances for each method where the best objective function value was found among the 375 instances.

Solution Method	**Number of Instances**	**Total**
Mixed Integer Linear Programming Model	140	140
Greedy Heuristic v. 1	2	
Greedy Heuristic v. 2	2	
Greedy Heuristic v. 3	2	
Greedy Heuristic v. 4	0	165
Greedy Heuristic v. 5 (10)	32	
Greedy Heuristic v. 5 (20)	41	
Greedy Heuristic v. 5 (40)	42	
Greedy Heuristic v. 5 (50)	44	
Tabu Search Configuration 1	27	
Tabu Search Configuration 2	2	
Tabu Search Configuration 3	65	
Tabu Search Configuration 4	1	
Tabu Search Configuration 5	0	126
Tabu Search Configuration 6	14	
Tabu Search Configuration 7	0	
Tabu Search Configuration 8	11	
Tabu Search Configuration 9	6	

Table 8.1: Summary of results achieved from the optimisation models and algorithms presented in this thesis. The number of instances where the best result known was found is shown.

8.3 Future Work

In this section a series of ideas for future work are described, firstly with regards to the WSRP definition and how it could be extended. Secondly a description of possible solution methods that can be used that were not considered in this thesis but that might produce good outcomes. Thirdly the problem of implementation is addressed, perhaps using a different implementation of both the greedy heuristic and the TS could increase the quality of results obtained.

In terms of the workforce scheduling and routing problem description, a clear continuation of work is to consider the case of multiple working days in the planning horizon. Further research in such a direction would have to incorporate Rostering type constraints, i.e. if an employee works two night shifts in a row, he is not eligible for a third night shift. Other rostering constraints will depend on the sector for which the problem is applied, i.e. the requirements of nurse rostering are not the same as those for lorry driver rostering. Another work extension could be focusing on producing balanced/fairer schedules for employees. In fact, there are already some publications related to medium to long term planning of home health care scheduling and routing (Matta et al., 2014; Carello and Lanzarone, 2014; Cappanera and Scutella, 2014)

In terms of the solution methods for tackling WSRP, other approaches should be explored. These include: constraint programming (Rendl et al., 2012) and column generation (Trautsamwieser and Hirsch, 2014). The rationale is that WSRPs can be heavily constrained due to the activities' time windows and the time-dependent constraints. A great deal of the effort spent on this research programme was in maintaining feasibility by satisfying the constraints. This makes a strong case for the use of constraint programming approaches. Column generation, i.e. branch and price, has proven to be a successful method for tackling huge mixed-integer models such as the MIP model version of WSRP. Another method could include an hybridisation approach (Di Gaspero and Urli, 2014; Masmoudi and Mellouli, 2014) such as Matheuristics which combine aspects of mathematical programming and heuristics methods. Finally, a range of methods that has not been included in this thesis but has also reported good results in similar problems, such as the variants of the VRP, is Greedy Randomised Adaptive Search Procedure GRASP (Ait Haddadene et al., 2014). In fact the different version of the Greedy Heuristic could be translated into a GRASP algorithm. GRASP is a multi-start metaheuristic which combines two stages: a construction and a local search one (Resende and Ribeiro, 2003). The construction stage build a feasible solution that then is passed to the local search component to evaluate further. The greedy heuristic could be considered the construction stage and then apply a local search the TS for example. At the end of Chapter 7 the manual setting of parameters based on the results of previous configuration could be included in a learning mechanism i.e. Reactive GRASP. The multi-start characteristic of GRASP and other heuristics could benefit of a parallel approach.

The major benefit of implementing a parallel approach is to speed the search procedure allowing multiple processors to work at the same time in parts of the algorithm that do not need to run in a serial manner. According to Crainic and Toulouse (2003) there are three different types of parallelism in heuristic methods. The first type is used in concurrent execution of the operations within the algorithms. As a result, the last version of the Greedy Heuristic (GH5) could benefit of this type since many possible candidate solutions could be evaluated concurrently. The second type refers to the decomposition of the decision variables often implemented in a master-slave framework. Early work using decomposition for WSRP seems to be promising already (Laesanklang et al., 2015). Finally, the third type of parallelism refers to executing concurrent heuristics which might not be the same, or perhaps the same, but with different initialisation parameters. This last type could be used in the Greedy Heuristic to start different versions at the same time or even the Tabu Search using different configurations.

With reference to continuing the work started with the greedy heuristics, possible improvements can be made if incorporating a backtracking mechanism which supports more than just the two levels used in the branching version of the heuristic (GH5). Incorporating backtracking might contradict the notion of greediness though. GH5 when performing branching was only able to explore two levels forward and then was forced to choose the best improving solution whilst discarding the other options, otherwise it risks running out memory. A parallel approach already discussed in the previous paragraph could allow multiple processes to continue the search with the discarded options, which might find better solutions as the searching process progresses.

Regarding the TS implementation, there are three options for future work. The first option is updating the swap and shift move types to handle activities with time-dependent constraints. As the results from Chapter 7 showed, the insertcca move seems to influence the most with regards to the quality of the results obtained. This could be explained because insertcca is the only move that affects time-dependent constraints. Therefore, extending the support for such constraints to other moves may also help to obtain better results. The second option is the addition of repair moves that could be included in order to allow infeasible solutions during the search. The repair moves will try to change the infeasible solution to a valid one. Another application of repair moves is that they could be applied to a current valid solution that suddenly has to be changed due to some external factor, the moves will then try to correct the solution ideally with minimal disruption. The third option for future work includes incorporating a learning mechanism which allow the TS to update some of the parameters that are static, e.g. adapting/tuning the precision parameters for the moves at different stages of the search. The introduction of a learning mechanism could lead to the usage of other high level methods such as Hyper-heuristics. Hyper-heuristics are applicable to many domain problems, as they separate the domain information from the search process. They are adaptive methods, since they chose at any time during the search process the best heuristic to use depending on the state of the search (Burke et al., 2003). Hyper-heuristics generate an online score of the performance of the low-level heuristics chosen by the heuristic selection method, which in many cases is a metaheuristic. All scored based selection techniques require five components: initial scoring, memory adjustment, strategy for selection, score update rules for improvement and worsening. Another type of Hyper-heuristics create heuristics based on low-level components and once generated the new heuristics prevail for future use and participate in the generation of others (Burke et al., 2013).

Finally, the majority of the work for in this thesis was performed using a weighted sum for the objective function that heavily favoured the assignment of activities over employees' preferences or cost. However, given the difference of the sectors where WSRP can be applied, e.g. home care, retail, service industry, etc. the case for devoting research to the multi-objective nature of the problem also seems to be a sound direction to progress this research.

Bibliography

Ait Haddadene, S. R., Labadie, N., and Prodhon, C. (2014). Grasp for the vehicle routing problem with time windiws, synchonization and precedence constraints. In *Proceeding of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications.*

Akjiratikarl, C., Yenradee, P., and Drake, P. R. (2006). An improved particle swarm optimization algorithm for care worker scheduling. In *Proceedings of the 7th Asia Pacific Industrial Engineering and Management Systems Conference 2006*, pages 457–466.

Akjiratikarl, C., Yenradee, P., and Drake, P. R. (2007). Pso-based algorithm for home care worker scheduling in the uk. *Computers & Industrial Engineering*, 53(4):559–583.

Alfares, H. K. (2004). Survey, categorization and comparison of recent tour scheduling literature. *Annals of Operations Research*, 127(1–4):145–175.

Allaoua, H., Sylvie, B., Létocart, L., and Wolfler Calvo, R. (2013). A matheuristic approach for solving a home health care problem. *Electronic note in Discrete Mathematics*, 41:471–478.

An, Y.-J., Kim, Y.-D., Jeong, B., and Kim, S.-D. (2012). Scheduling healthcare services in a home healthcare system. *Journal of the Operational Research Society*, 63:1589–1599.

Azi, N., Gendreau, M., and Potvin, J.-Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, 202(3):756–763.

Baker, K. R. (1976). Workforce allocation in cyclical scheduling problems: A survey. *Journal of the Operational Research Society*, 27(1):155–167.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329.

Bechtold, S., Brusco, M. J., and Showalter, M. (1991). A comparative evaluation of labour tour scheduling methods. *Decision Science*, 22(4):683–699.

Begur, S. V., Miller, D. M., and Weaver, J. R. (1997). An integrated spatial dss for scheduling and routing home-health-care nurses. *Interfaces*, 27(4):35–48.

Bertels, S. and Fahle, T. (2006). A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. *Computers & Operations Research*, 33(10):2866–2890.

Blais, M., Lapierre, S. D., and Laporte, G. (2003). Solving a home-care districting problem in a urban setting. *Journal of the Operational Research Society*, 54(11):1141–1147.

Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308.

Borsani, V., Matta, A., Sommaruga, F., and Beschi, G. (2006). A home care scheduling model for human resources. In *Service Systems and Service Management, 2006 International Conference on*, volume 1, pages 449–454.

Bostel, N., Dejax, P., Guez, P., and Tricoire, F. (2008). Multiperiod planning and routing on a rolling horizon for field force optimization logistics. In Golden, B., Raghavan, S., and Wasil, E., editors, *The Vehicle Routing Problem: Last Advances and New Challanges*, pages 503–525. Springer US.

Bozkaya, B., Erkut, E., and Laporte, G. (2003). A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1):12–26.

Brandão, J. and Mercer, A. (1997). A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research*, 100(1):180–191.

Brandão, J. and Mercer, A. (1998). The multi-trip vehicule routing problem. *Journal of the Operational Research Society*, 49(8):799–805.

Bredström, D. and Rönnqvist, M. (2007). A branch and price algorithm for the combined vehicle routing and scheduling problem with synchronization constraints. Technical report, NHH Dept. of Finance & Management Science Discussion Paper No. 2227/7.

Bredström, D. and Rönnqvist, M. (2008). Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research*, 191(1):19–31.

Brucker, P. and Knust, S. (2006). *Complex Scheduling*, chapter 2, pages 23–90. Springer.

Brucker, P., Qu, R., and Burke, E. (2011). Personned scheduling: Mmodel and complexity. *European Journal of Operational Research*, 210(3):467–473.

Burke, E., Gendreau, M., M., H., Kendall, G., G., O., Ozcan, E., and Qu, R. (2013). Hyper-heuristics: a curvey of the state of the art. *Journal of the Operational Research Society*, 64.

Burke, E., Kendall, G., Newal, J., Hart, E., Ross, P., and Schulenburg, S. (2003). *Hyper-Heuristics: An emerging direction in modern search technologies*, chapter 16, pages 457–474. Kluwers Academic Publising.

Burke, E. K. and Kendall, G. (2005). *Search Methodologies: Introductory tutorial in optimization and desicion support techniques*, chapter 1, pages 5–18. Springer, 1st edition.

Burke, E. K. and Kendall, G. (2014). *Search Methodologies: Introductory tutorial in optimization and desicion support techniques*, chapter 1, pages 5–18. Springer, 1st edition.

Cai, Y., Zhang, Z., Guo, S., Qin, H., and Lim, A. (2013). A tree-based tabu search algorithm for the manpower allocation problem with time windows and job-teaming constraints. In *Proceedings of theTwenty-Third international joint conference on Artificial Intelligence*.

Cappanera, P. and Scutella, M. G. (2014). Joint assignment, scheduling, and routing modles to home care optimization: A pattern-based approach. *Transportation Science*.

Carello, G. and Lanzarone, E. (2014). A cardinality-constrained robust model for the assignment problem in home care services. *European Journal of Operational Research*, 236(2):748–762.

Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2012). A survey on workforce scheduling and routing problems. In *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)*, pages 283–302, Son, Norway.

Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2014a). Computational study for workforce scheduling and routing problems. In *ICORES 2014 - Proceedings of the 3rd International Conference on Operations Research and Enterprise Systems*, pages 434–444.

Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2014b). Workforce scheduling and routing problems: literature survey and computational study. *Annals of Operations Research*, doi: 10.1007/s10479-014-1687-2.

Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2015). A greedy heuristic for workforce scheduling and routing with time-dependent activities constraints. In *ICORES 2015 - Proceedings of the 4th International Conference on Operations Research and Enterprise Systems*, pages 367–375, Lisbon, Portugal. INSTICC, Scitepress.

Castro-Gutierrez, J., Landa-Silva, D., and Moreno-Perez, J. (2011). Nature of real-world multi-objective vehicle routing with evolutionary algorithms. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 257 –264.

Cheang, B., Li, H., Lim, A., and Rodrigues, B. (2003). Nurse rostering problems–a bibliographic survey. *European Journal of Operational Research*, 151(3):447–460.

Cheng, E. and Rich, J. L. (1998). A home health care routing and scheduling problem.

Chuin Lau, H. and Gunawan, A. (2012). The patrol scheduling problem. In *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)*, pages 175–192, Son, Norway.

Cordeau, J.-F. and Laporte, G. (2003). The dial-a-ride problem (darp): Variants, modeling issues and algorithms. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(2):89–101.

Cordeau, J.-F., Laporte, G., and Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Jornal of the Oper, Journal of the Operational Research Society*, 52:928–936.

Cordeau, J.-F., Laporte, G., Pasin, F., and Ropke, S. (2010). Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, 13(4):393–409.

Cortés, C. E., Gendreau, M., Rousseau, L. M., Souyris, S., and Weintraub, A. (2014). Branch-and-price and constraint programming for solving a real-life technician dispatching problem. *European Journal of Operational Research*, 238:300–312.

Crainic, T. G. and Toulouse, M. (2003). *Parallel Strategies for Meta-heuristics*, chapter 17, pages 475–513. Kluwers Academic Publising.

Danna, E. and Le Pape, C. (2005). Branch-and-price heuristics: A case study on the vehicle routing problem with time windows. In Dasaulniers, G., Desrosiers, J., and

Solomon, M. M., editors, *Column Generation*, chapter 4, pages 99–129. Springer US.

Dawande, M., Kalagnanam, J., Keskinocak, P., Salman, F., and Ravi, R. (2000). Approximation aalgorithmsf the multiple knapsack problem with assignment restrictions. *Journal of Combinatorial Optimization*, 4(2):171–186.

De Angelis, V. (1998). Planning home assistance for aids patients in the city of rome, italy. *Interfaces*, 28(3):75–83.

Desaulniers, G., Lavigne, J., and Soumis, F. (1998). Multi-depot vehicle scheduling problems with time windows and waiting costs. *European Journal of Operational Research*, 111(3):479–494.

Desrochers, M., Desrosiers, J., and Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2):342–354.

Desrochers, M., Lenstra, J. K., and Savelsbergh, M. W. P. a. (1990). A classification scheme for vehicle routing and scheduling problems. *European Journal of Operational Research*, 46(3):322–332.

Desrosiers, J. and Lübbecke, M. E. (2005). A primer in column generation. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, chapter 1, pages 1–32. Springer.

Devarenne, I., Mabed, H., and Caminada, A. (2008). *Adaptive Tabu Tenure Computation in Local Search*, chapter 1, pages 1–12. Springer Berlin Heidelberg.

Di Gaspero, L. and Urli, T. (2014). A cp/lns approach for multi-day homehome schedulling problems. In *Hybrid Metaheuristics*, chapter 1, pages 1–15. Springer International Publishing.

Doerner, K. F. and Hartl, R. F. (2008). Health care logistics, emergency preparedness, and disaster relief: New challenges for routing problems with a focus on the austrian situation. In Golden, B., Raghavan, S., and Wasil, E., editors, *The Vehicle Routing Problem: Last Advances and New Challanges*, pages 527–550. Springer US.

Dohn, A., Kolind, E., and Clausen, J. (2009). The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach. *Computers & Operations Research*, 36(4):1145–1157.

Dohn, A., Rasmussen, M. S., Justesen, T., and Larsen, J. (2008). The home care crew scheduling problem. In Sheibani, K., editor, *Proceedings of the 1st International*

Conference on Applied Operational Research, volume 1 of *Lecture Notes in Management Science*, pages 1–8. Institute for Operational Research, System Design and Financial Services, Tadbir.

Dohn, A., Rasmussen, M. S., and Larsen, J. (2011). The vehicle routing problem with time windows and temporal dependencies. *Networks*, 58:273–289.

Dorigo, M. and Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66.

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). Ant system: optimization by a colony of cooperative agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 26(1):29–41.

Dowsland, K. A. (2014). *Classical Techniques*, chapter 2, pages 19–65. Springer, 2nd edition.

Eberhart, R. C. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceeedings of the sixth international symposium on micro machine and human science*, pages 39–43.

Ernst, A., Jiang, H., Krishnamoorthy, M., Owens, B., and Sier, D. (2004a). An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127:21–144.

Ernst, A., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004b). Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, 153(1):3–27.

Eveborn, P., Flisberg, P., and Rönnqvist, M. (2006). Laps care – an operational system for staff planning of home care. *European Journal of Operational Research*, 171(3):962–976.

Eveborn, P., Rönnqvist, M., Einarsdóttir, H., Eklund, M., Lidén, K., and Almroth, M. (2009). Operations research improves quality and efficiency in home care. *Interfaces*, 39(1):18–34.

Feillet, D. (2010). A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR: A Quarterly Journal of Operations Research*, 8(4):407–424.

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Applications to some vehicle routing problems. *Networks*, 44(3):216–229.

Firat, M. and Hurkens, C. A. J. (2012). An improved mip-based approach for a multi-skill workforce scheduling problem. *Journal of Scheduling*, 15(3):363–380.

Fischetti, M., Polo, C., and Scantamburlo, M. (2004). A local branching heuristic for mixed-integer programs with 2-level variables, with an application to a telecommunication network design problem. *Networks*, 44(2):61–72.

Fukasawa, R., Lysgaard, J., Poggi de Aragao, M., Reis, M., Uchoa, E., and Werneck, R. F. (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3):491–511.

Gendreau, M. and Potvin, J.-Y. (2014). Tabu search. In Burke, E. K. and Kendall, G., editors, *Search Methodologies Introductory Tutorials in OPtimization and Decision Support Techniques*, chapter 9, pages 243–263. Springer.

Gendreau, M., Potvin, J.-Y., Bräysy, O., Hasle, G., and Lokketangen, A. (2008). Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. In *The Vehicle Routing Problem Latest Advances and New Challenges*, pages 143–169. Springer.

Glover, F. (1989). Tabu search–part 1. *ORSA Journal on Computing*, 1(3):190–206.

Glover, F. (1990a). Tabu search–part ii. *ORSA Journal on Computing*, 2(1):4–32.

Glover, F. (1990b). Tabu search: A tutorial. *Interfaces*, 20(4):74–94.

Glover, F. and Laguna, M. (1999). *Tabu Search*. Springer US.

Golden, B., Raghavan, S., and Wasil, E., editors (2008). *The Vehicle Routing Problem: Latest advances and new challenges*. Springer.

Golembiewski, R. and Proehl Jr, C. (1978). A survey of the empirical literature on flexible workhours: Character and consequences of a major innovation. *Academy of Management Review*, 3(4):837–853.

Günther, M. and Nissen, V. (2012). Application of particle swarm optimization to the british telecom workforce scheduling problem. In *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)*, pages 242–256, Son, Norway.

Halvorsen-Weare, E. and Fagerholt, K. (2013). Routing and scheduling in a liquefied natural gas shipping problem with inventory and berth constraints. *Annals of Operations Research*, 203:167–186.

Harder, R. (2004). *OpenTS Tutorial - Traveling Salesman Problem*.

Harder, R. W., Hill, R. R., and Moore, J. (2004). A java universal vehicle router for routing unmanned aerial vehicles. *International Transactions in Operational Research*, 11:259–275.

Hashimoto, H., Boussier, S., Vasquez, M., and Wilbaut, C. (2011). A grasp-based approach for technicians and interventions scheduling for telecommunications. *Annals of Operations Research*, 183:143–161.

Hillier, F. S. and Lieberman, G. J. (2010). *Introduction to Operations Research*. McGraw-Hill.

Inc., G. O. (2013). *Gurobi Optimizer, Reference Manual version 5.5*. Gurobi Optimization Inc.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, chapter 2, pages 33–65. Springer.

Itabashi, G., Chiba, M., Takahashi, K., and Kato, Y. (2006). A support system for home care service based on multi-agent system. In *Information, Communications and Signal Processing 2005 Fifth International Conference on*, pages 1052–1056.

Justesen, T. and Rasmussen, M. S. (2008). The home care crew scheduling problem. Master's thesis, Informatics and Mathematical Modeling, Technical University of Denmark Department of Computer Science, University of Copenhagen.

Kallehauge, B., Larsen, J., Madsen, O. B. G., and Solomon, M. M. (2005). Vehicle routing problem with time windows. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, chapter 3, pages 67–98. Springer.

Kergosien, Y., Lente, C., and Billaut, J.-C. (2009). Home health care problem, an extended multiple travelling salesman problem. In Blazewicz, J., Drozdowski, M., Kendall, G., and McCollum, B., editors, *Proceedings of the 4th Multidisciplinary International Scheduling Conference: Theory and Applications (MISTA 2009)*, pages 85–92.

Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5/6):671–680.

Korsah, G. A., Stentz, A., Dias, M. B., and Fanaswala, I. (2010). Optimal vehicle routing and scheduling with precedence constraints and location choice.

Kovacs, A. A., Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2012). Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of Scheduling*, 15:579–600.

Laesanklang, W., Landa-Silva, D., and Castillo-Salazar, J. A. (2015). Mixed integer programming with decomposition to solve a workforce scheduling and routing problem. In *ICORES 2015 - Proceedings of the 4th International Conference on Operations Research and Enterprise Systems*, pages 283–293, Lisbon, Portugal. INSTICC, Scitepress.

Landa-Silva, D., Wang, Y., Donovan, P., and Kendall, G. (2011). Hybrid heuristic for multi-carrier transportation plans. In *Proceedins of the 9th Metaheuristics International Conference (MIC2011)*, pages 221–229.

Lawler, E. L. and Wood, D. E. (1966). Branch-and-bound methods: A survey. *Operations Research*, 14(4):699–719.

Lenstra, J. K. and Rinnooy Kan, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11:221–227.

Lesaint, D., Voudouris, C., Azarmi, N., Alletson, I., and Laithwaite, B. (2003). Field workforce scheduling. *BT Technology Journal*, 21(4):23–26.

Li, Y., Lim, A., and Rodrigues, B. (2005). Manpower allocation with time windows and job-teaming constraints. *Naval Research Logistics*, 52(4):302–311.

Lim, A., Rodrigues, B., and Song, L. (2004). Manpower allocation with time windows. *Journal of the Operational Research Society*, 55:1178–1186.

Mankowska, D. S., Meisel, F., and Bierwirth, C. (2014). The home health care routing and scheduling problem with interdependent services. *Health Care Management Science*, 17:15–30.

Martin, A. (2001). General mixed integer programming: Computational issues for branch-and-cut algorithms. In Jünger, M. and Naddef, D., editors, *Computational Combinatorial Optimization*, Lecture Notes in Computer Science, chapter 1, pages 1–25. Springer Berlin Heidelberg.

Masmoudi, M. and Mellouli, R. (2014). Milp for synchronized-mtsptw: application to home healhealth scheduling. In *Control, Decision and Information Technologies (CoDIT)*.

Matta, A., Chahed, S., Sahin, E., and Dallery, Y. (2014). Modemodel home care organisations from and operations management perspective. *Flexible Services and Manufacturing Journal*, 26(3):295–319.

Miller, H. (1976). Personnel scheduling in public systems: a survey. *Socio-economic planning sciences*, 10(6):241–249.

Misir, M., Smet, P., and Vanden Bergue, G. (2015). An analysis of generalised heuristics for vehicle routing and personnel rostering problems. *Journal of the Operational Research Society*, 66(5):858–870.

Misir, M., Smet, P., Verbeeck, K., and Vanden Bergue, G. (2011). Security personnel routing and rostering: a hyper-heuristic approach. In *Proceedings of the 3rd International Conference on Applied Operational Research, ICAOR11, Istanbul, Turkey, 2011*, pages 193–206.

Misir, M., Verbeeck, K., De Causmaecker, P., and Vanden Bergue, G. (2010). Hyper-heuristics with a dynamic heuristic set for the home care scheduling problem. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 18–23.

Mitchell, J. E. (2002). *Branch-and-cut algorithms for combinatorial optimization problems*, chapter 3, pages 53–65. Oxford University Press.

Mitten, L. G. (1970). Branch-and-bound methods: General formulation and properties. *Operations Research*, 18(1):24–34.

Nickel, S., Schröder, M., and Steeg, J. (2012). Mid–term abd short–term planning support for home health care services. *European Journal of Operational Research*, 219:574–587.

Osman, I. H. and Laporte, G. (1996). Metaheuristics: A bibliography. *Annals of Operations Research*, 63:513–623.

Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal fr Betriebswirtschaft*, 58(1):21–51.

Pillac, V., Guéret, C., and Medaglia, A. L. (2011). On the technician routing and scheduling problem. In *Proceedings of the IX Metaheuristics International Conference (MIC 2011), Udine, Italy, July 25–28*.

Pillac, V., Guéret, C., and Medaglia, A. L. (2013). A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, 7:1525–1535.

Pinedo, M. L. (2009). *Planning and Scheduling in Manufacturing and Services*, chapter 13, pages 317–343. Springer-Verlag New York.

Polacek, M., Hartl, R. F., Doerner, K., and Reimann, M. (2004). A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics*, 10(6):613–627.

Potvin, J.-Y. and Rousseau, L. M. (1995). An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society*, 46:1433–1446.

Raff, S. (1983). Routing and scheduling of vehicles and crews: The state of the art. *Computers & Operations Research*, 10(2):63–211.

Rasmussen, M. S., Justesen, T., Dohn, A., and Larsen, J. (2012). The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, 219(3):598–610.

Rees, C. R. (1996). Modern heuristic techniques. In Rayward-Smith, V. J., Osman, I. H., Reeves, C. R., and Smith, G. D., editors, *Modern Heuristic Search Methods*, pages 1–25. Wiley, New York.

Rendl, A., Prandtstetter, M., Hiermann, G., Puchinger, J., and Raidl, G. (2012). Hybrid heuristics for multimodal homehome scheduling. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems.*

Resende, M. G. and Ribeiro, C. C. (2003). *Greedy Randomized Adaptive Search Procedures*, chapter 8, pages 219–249. Kluwers Academic Publising.

Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.

Salani, M. and Vaca, I. (2011). Branch and price for the vehicle routing problem with discrete split deliverables and time windows. *European Journal of Operational Research*, 213(3):470–477.

Schwerin, P. and G., W. (1997). The bin-packing problem: A problem generator and some numerical experiments with ffd packing and mtp. *International Transactions in Operational Research*, 4(5–6):377–389.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time windows constraints. *Operations Research*, 35(2):254–265.

Stützle, T. and Hoos, H. H. (2000). Max-min ant system. *Future Generation Computer System*, 16(8):889–914.

Talbi, E.-G. (2009). *Metaheuristics From Design to Implementation.* John Wiley & Sons.

Thomsen, K. (2006). Optimization on home care. Master's thesis, Department of informatics and Mathematical Modelling (IMM), Technical University of Denmark (DTU).

Toth, P. and Vigo, D. (1987). *The vehicle routing problem*, volume 9. Society for Industrial and Applied Mathematics.

Trautsamwieser, A. and Hirsch, P. (2011). Optimization of daily scheduling for home health care services. *Journal of Applied Operational Research*, 3(3):124–136.

Trautsamwieser, A. and Hirsch, P. (2014). A branch-price-and-cut approach for solving the medium-term home health care planning problem. *Networks*, 64:143–159.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research*, 226:367–385.

Voß, S., Martello, S., Osman, I. H., and Roucairol, C., editors (1999). *Mete-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer Academic Publishers.

Ward Athan, T. and Papalambros, P. Y. (1996). A note on weighted criteria methods for compromise ssolution in multi-objective optimization. *Engineering Optimization*, 27(2):155–176.

Weigel, D. and Cao, B. (1999). Applying gis and or techniques to solve sears technician-dispatching and home-delivery problems. *Interfaces*, 29(1):112–130.

Xu, J. and Chiu, S. Y. (2001). Effective heuristic procedures for a field technician scheduling problem. *Journal of Scheduling*, 7:495–509.

Appendix A

Data set summary

Table A.1: Shows for each instance information regarding number of employees (Emp), employees's coverage of activities based on skills (Skill), number of activities (Act), mean activity duration (μ Act), mean time window duration (μ TW), planning horizon duration (PH) and number of time-dependent constraints (T.D.C.). In the last column the time-dependent constraints are ordered as follow: synchronisation, overlapping, minimum time difference, maximum time difference and min-max time difference.

Instance	Emp	Skill	Act	μ Act	μ TW	PH	T.D.C
10_District0	13	0.83	52	282.12	415.38	1440	2,0,3,0,2
10_District1	29	0.96	118	531.48	501.47	1440	3,5,4,4,6
10_District2	13	0.95	52	368.94	452.71	1440	1,1,0,2,2
10_District3	14	1.	58	384.57	405.	1440	1,1,2,2,2
10_District4	39	0.97	159	425.38	485.89	1440	2,2,6,5,5
10_District5	39	0.96	156	532.5	496.58	1440	1,5,5,7,6
11_District0	7	0.7	31	298.06	498.39	1440	2,0,3,1,1
11_District1	28	0.99	112	529.82	494.59	1440	2,4,1,4,10
11_District2	9	0.95	37	387.57	527.81	1440	0,1,3,1,0
11_District3	13	1.	53	332.55	485.09	1440	4,0,2,2,2
11_District4	38	0.94	153	477.45	487.99	1440	1,3,4,6,6
11_District5	34	0.91	136	522.46	423.63	1440	3,0,9,4,6
12_District0	14	0.47	57	296.84	362.11	1440	1,3,1,3,1
12_District1	41	0.97	164	492.8	499.1	1440	6,1,8,9,5
12_District2	15	0.83	61	369.34	435.11	1440	4,0,3,1,0
12_District3	17	0.97	71	334.44	428.03	1440	4,1,3,1,4
12_District4	47	0.92	190	434.76	514.06	1440	5,5,6,4,8
12_District5	46	0.95	187	508.07	435.8	1440	4,8,6,7,6
13_District0	14	0.43	58	317.07	457.76	1440	2,2,0,1,2
13_District1	34	0.99	138	499.46	519.22	1440	2,5,8,4,3
13_District2	16	0.94	64	316.17	387.63	1440	2,0,4,5,1
13_District3	19	0.99	78	345.77	452.68	1440	4,3,2,4,5
13_District4	40	0.9	161	435.47	519.27	1440	10,2,11,6,9
13_District5	42	0.94	168	464.11	457.76	1440	5,5,11,8,4
14_District0	11	0.45	44	268.64	448.64	1440	4,2,3,1,0
14_District1	33	1.	134	477.09	526.66	1440	4,4,6,8,4
14_District2	13	0.89	55	396.82	456.4	1440	4,2,3,2,1
14_District3	17	0.99	71	343.52	430.24	1440	3,2,1,4,2
14_District4	41	0.95	167	468.14	509.86	1440	6,6,3,7,8
14_District5	42	0.94	169	486.04	429.75	1440	6,6,6,10,9

Table A.1 – continued from previous page

Instance	Emp	Skill	Act	μ Act	μ TW	PH	T.D.C
15_District0	12	0.58	51	258.24	410.29	1440	3,4,1,0,2
15_District1	32	0.97	130	511.85	499.95	1440	6,4,3,5,7
15_District2	14	0.87	58	334.4	375.5	1440	4,1,4,2,0
15_District3	19	1.	78	320.58	388.27	1440	4,1,6,6,1
15_District4	40	0.95	163	413.37	519.28	1440	4,2,2,6,7
15_District5	39	0.98	157	495.38	477.89	1440	3,4,3,6,5
16_District0	13	0.38	54	287.78	425.56	1440	3,4,2,2,0
16_District1	32	0.93	128	506.13	448.12	1440	6,1,6,2,2
16_District2	13	0.81	54	424.44	441.24	1440	2,1,3,1,3
16_District3	17	0.96	68	344.12	387.46	1440	4,2,2,1,4
16_District4	41	0.92	166	443.58	516.63	1440	3,4,5,9,9
16_District5	43	0.94	175	485.14	485.86	1440	6,5,8,2,6
17_District0	12	0.42	48	312.5	448.75	1440	3,6,2,1,0
17_District1	31	0.93	125	532.44	498.05	1440	4,6,2,4,3
17_District2	14	0.95	58	391.55	478.19	1440	2,3,3,4,0
17_District3	15	1.	63	361.9	397.97	1440	4,2,3,2,1
17_District4	38	0.98	155	421.35	467.46	1440	5,5,5,3,7
17_District5	36	0.98	146	544.42	475.63	1440	1,3,8,8,4
18_District0	6	0.51	26	267.69	399.23	1440	0,0,1,2,2
18_District1	31	0.92	127	526.77	502.38	1440	6,3,5,6,7
18_District2	8	0.85	33	331.36	377.03	1440	2,1,0,0,1
18_District3	12	1.	49	396.12	475.41	1440	2,0,1,3,4
18_District4	36	0.95	147	435.41	514.48	1440	5,8,9,4,5
18_District5	31	0.98	126	544.88	492.79	1440	2,6,6,4,8
19_District0	12	0.66	49	268.16	427.35	1440	2,0,4,2,0
19_District1	36	0.97	146	506.71	536.45	1440	3,7,4,7,6
19_District2	15	0.86	62	394.35	490.52	1440	5,2,1,2,0
19_District3	17	0.92	69	325.87	436.41	1440	0,4,6,3,4
19_District4	52	0.94	210	443.07	535.21	1440	5,7,8,11,7
19_District5	47	0.88	191	503.09	441.56	1440	7,7,10,9,10
1_District0	18	0.6	73	303.7	427.6	1440	6,2,4,6,0
1_District1	44	0.93	176	490.65	539.47	1440	6,8,10,4,9
1_District2	19	0.8	78	390.19	500.27	1440	4,1,5,4,2
1_District3	22	0.95	90	341.5	400.	1440	2,5,2,2,3
1_District4	51	0.98	204	431.1	495.33	1440	6,3,10,3,9
1_District5	49	0.96	197	489.14	469.41	1440	3,8,11,9,6
20_District0	13	0.39	53	290.38	392.83	1440	2,3,3,3,1
20_District1	33	0.88	135	511.33	509.44	1440	1,5,1,3,7
20_District2	12	0.96	49	353.27	442.65	1440	3,1,3,1,3
20_District3	18	0.99	74	297.97	408.84	1440	2,3,3,1,6
20_District4	41	0.91	165	418.73	537.35	1440	3,3,4,5,5
20_District5	44	0.93	178	471.74	455.26	1440	7,2,9,5,11
21_District0	15	0.65	60	265.	407.75	1440	2,1,7,2,1
21_District1	34	0.99	139	495.	477.45	1440	2,5,7,6,8
21_District2	13	0.98	55	402.	569.44	1440	2,1,1,1,0
21_District3	21	0.99	84	378.21	478.75	1440	1,0,4,1,4
21_District4	39	0.96	159	456.6	552.64	1440	4,2,10,7,8
21_District5	42	0.93	171	490.7	482.4	1440	4,4,6,7,4
22_District0	14	0.5	56	297.32	454.02	1440	1,2,2,4,6
22_District1	33	1.	132	494.77	449.71	1440	4,4,4,7,4
22_District2	16	0.91	65	359.31	417.09	1440	1,2,1,3,1
22_District3	18	1.	75	327.6	370.8	1440	4,1,5,3,4
22_District4	40	0.89	162	441.94	498.28	1440	3,4,3,11,8
22_District5	41	0.98	165	505.45	485.45	1440	1,3,7,5,8
23_District0	10	0.32	42	292.86	412.86	1440	2,0,3,0,3
23_District1	35	0.96	141	503.72	484.36	1440	3,6,7,4,4

Table A.1 – continued from previous page

Instance	Emp	Skill	Act	μ Act	μ TW	PH	T.D.C
23_District2	15	0.78	62	386.85	412.35	1440	6,0,3,3,2
23_District3	17	0.99	69	381.3	415.43	1440	1,3,2,2,3
23_District4	42	0.97	168	452.41	493.39	1440	7,6,4,7,5
23_District5	46	0.98	186	495.73	444.99	1440	9,5,4,6,4
24_District0	14	0.5	56	305.36	418.13	1440	3,1,1,5,2
24_District1	31	0.95	126	535.6	466.72	1440	8,2,6,3,5
24_District2	10	0.88	43	392.09	351.26	1440	0,0,2,1,3
24_District3	16	1.	64	345.	480.11	1440	2,3,4,1,3
24_District4	41	0.96	165	429.09	512.81	1440	5,9,6,10,4
24_District5	36	0.95	145	542.28	461.27	1440	5,6,2,5,7
25_District0	6	0.81	26	286.15	437.31	1500	1,1,2,1,2
25_District1	28	1.	113	530.44	529.57	1500	5,0,4,4,6
25_District2	7	0.86	28	358.39	406.61	1500	1,1,1,1,0
25_District3	11	0.98	46	415.43	553.37	1500	2,1,0,2,1
25_District4	38	0.96	154	448.15	523.25	1500	4,3,7,9,7
25_District5	30	0.97	123	565.61	519.94	1500	2,3,5,6,2
26_District0	16	0.66	65	296.77	416.54	1440	3,3,4,4,3
26_District1	41	0.96	164	506.34	474.51	1440	2,2,9,11,1
26_District2	17	0.82	69	378.7	437.71	1440	2,2,4,6,2
26_District3	23	0.98	92	352.83	423.75	1440	5,3,7,1,1
26_District4	52	0.95	210	462.14	513.28	1440	4,2,10,16,9
26_District5	48	0.98	193	485.98	496.85	1440	6,3,10,8,8
27_District0	15	0.58	60	248.5	362.	1440	3,3,1,3,4
27_District1	34	0.94	138	512.39	539.02	1440	1,6,4,4,12
27_District2	14	0.9	56	356.79	499.68	1440	1,4,0,3,2
27_District3	15	0.98	60	299.75	356.62	1440	2,2,0,4,4
27_District4	43	0.93	174	411.55	525.29	1440	3,4,6,7,10
27_District5	42	0.96	169	489.05	462.19	1440	4,5,4,9,5
28_District0	13	0.53	55	316.91	468.27	1440	1,1,1,4,2
28_District1	34	0.93	137	490.84	558.82	1440	1,2,7,4,5
28_District2	13	1.	52	385.96	361.27	1440	1,1,2,3,4
28_District3	21	1.	86	323.72	443.02	1440	5,4,1,1,7
28_District4	40	0.97	162	419.72	477.96	1440	5,2,7,6,5
28_District5	40	0.96	161	487.73	459.49	1440	3,2,8,5,9
29_District0	11	0.53	44	278.18	400.23	1440	2,1,0,2,2
29_District1	29	0.96	116	467.97	462.79	1440	4,8,4,3,6
29_District2	14	0.93	58	366.72	513.72	1440	4,1,4,2,2
29_District3	17	0.99	69	348.04	430.54	1440	1,4,2,1,5
29_District4	32	0.94	130	443.88	551.76	1440	5,5,5,0,5
29_District5	41	1.	166	478.46	457.35	1440	2,11,4,5,6
2_District0	15	0.72	60	274.5	402.75	1440	4,0,1,3,3
2_District1	40	0.96	163	507.42	456.02	1440	6,5,9,8,3
2_District2	16	0.91	66	348.86	426.45	1440	1,2,1,3,6
2_District3	21	0.98	86	350.76	452.35	1440	5,5,3,3,0
2_District4	47	0.95	190	454.11	503.68	1440	5,5,8,8,5
2_District5	50	0.96	201	492.54	458.01	1440	4,11,6,10,11
30_District0	9	0.44	38	285.	401.05	1440	1,2,2,3,2
30_District1	25	0.99	103	512.48	439.14	1440	5,4,5,6,3
30_District2	11	0.86	47	392.23	416.49	1440	1,1,2,1,2
30_District3	14	0.98	59	349.07	385.17	1440	5,0,2,2,2
30_District4	30	0.89	121	410.45	512.78	1440	6,5,4,3,8
30_District5	27	0.95	111	485.	484.78	1440	6,4,5,2,8
3_District0	13	0.61	54	296.11	429.17	1440	3,2,1,1,2
3_District1	33	0.96	135	532.11	546.27	1440	4,3,3,6,9
3_District2	16	0.91	64	389.77	402.88	1440	3,1,4,1,1
3_District3	21	1.	85	365.12	446.12	1440	3,2,2,5,2

Table A.1 – continued from previous page

Instance	Emp	Skill	Act	μ Act	μ TW	PH	T.D.C
3_District4	50	0.97	200	438.08	517.79	1440	2,4,7,5,9
3_District5	45	0.96	183	519.26	486.88	1440	6,3,8,6,8
4_District0	10	0.4	40	290.25	411.38	1440	1,4,0,1,1
4_District1	33	0.93	132	560.68	510.51	1440	4,4,3,2,4
4_District2	10	0.83	41	377.93	457.12	1440	2,0,1,0,3
4_District3	11	0.98	47	329.36	500.57	1440	3,4,0,1,2
4_District4	45	0.94	182	432.12	542.96	1440	5,7,7,7,7
4_District5	37	0.95	151	544.47	476.22	1440	6,4,5,8,5
5_District0	14	0.39	56	277.5	392.14	1440	3,3,2,1,2
5_District1	36	0.97	144	500.42	497.02	1440	1,2,7,5,15
5_District2	13	0.89	53	371.89	443.91	1440	2,5,1,2,3
5_District3	19	0.96	76	345.2	378.55	1440	3,5,3,2,3
5_District4	47	0.93	188	476.49	516.78	1440	1,7,4,6,9
5_District5	50	0.93	201	491.57	456.59	1440	7,3,7,7,10
6_District0	15	0.52	60	279.5	399.75	1440	1,3,4,3,0
6_District1	38	1.	155	463.16	458.94	1440	3,8,10,3,10
6_District2	17	0.82	70	341.14	435.41	1440	3,2,3,2,3
6_District3	16	1.	67	356.19	438.81	1440	2,2,2,2,4
6_District4	42	0.93	171	415.26	518.11	1440	2,7,11,6,6
6_District5	44	0.93	177	499.07	458.13	1440	3,5,10,7,8
7_District0	12	0.73	49	288.98	464.08	1440	2,1,0,3,0
7_District1	38	0.94	153	480.1	516.32	1440	6,4,4,3,5
7_District2	13	0.89	55	381.27	488.44	1440	2,4,3,3,1
7_District3	20	1.	82	323.23	395.12	1440	6,2,7,1,1
7_District4	41	0.91	167	443.08	484.75	1440	7,3,5,4,10
7_District5	45	0.96	181	469.97	471.66	1440	4,6,5,9,13
8_District0	12	0.54	49	274.29	425.51	1440	3,3,3,1,3
8_District1	32	0.98	130	533.65	485.18	1440	1,4,7,6,4
8_District2	13	0.71	53	341.89	371.89	1440	1,0,4,3,1
8_District3	18	1.	74	348.24	485.27	1440	5,1,4,4,5
8_District4	41	0.92	166	428.22	456.27	1440	7,7,6,3,5
8_District5	40	0.9	162	508.33	491.94	1440	4,7,9,4,7
9_District0	12	0.5	51	301.18	399.41	1440	0,2,1,2,3
9_District1	31	0.96	124	519.07	471.94	1440	3,3,5,3,5
9_District2	13	0.87	53	428.77	370.6	1440	1,4,2,2,1
9_District3	22	0.98	89	359.83	389.22	1440	2,5,5,2,3
9_District4	44	0.96	178	423.62	516.72	1440	6,3,9,5,8
9_District5	37	0.97	151	522.42	470.36	1440	1,5,4,9,6
C101_100t_20w	20	0.75	100	90.	60.76	1236	1,0,1,3,0
C101_25t_5w	5	0.6	25	90.	60.44	1236	0,0,0,1,0
C101_50t_10w	10	0.8	50	90.	60.14	1236	0,0,1,2,0
C102_100t_20w	20	0.75	100	90.	325.69	1236	7,4,1,3,0
C102_25t_5w	5	0.6	25	90.	359.44	1236	0,1,0,1,0
C102_50t_10w	10	0.8	50	90.	336.6	1236	0,5,1,2,0
C103_100t_20w	20	0.75	100	90.	588.49	1236	9,5,1,3,0
C103_25t_5w	5	0.6	25	90.	611.6	1236	0,2,0,1,0
C103_50t_10w	10	0.8	50	90.	590.44	1236	1,5,1,2,0
C104_100t_20w	20	0.75	100	90.	852.94	1236	11,5,1,3,0
C104_25t_5w	5	0.6	25	90.	781.48	1236	0,3,0,1,0
C104_50t_10w	10	0.8	50	90.	908.04	1236	1,5,1,2,0
C105_100t_20w	20	0.75	100	90.	121.61	1236	1,3,1,3,0
C105_25t_5w	5	0.6	25	90.	121.08	1236	0,2,0,1,0
C105_50t_10w	10	0.8	50	90.	120.38	1236	1,1,1,2,0
C106_100t_20w	20	0.75	100	90.	156.15	1236	1,1,1,3,0
C106_25t_5w	5	0.6	25	90.	73.72	1236	0,0,0,1,0
C106_50t_10w	10	0.8	50	90.	94.36	1236	0,0,1,2,0

Table A.1 – continued from previous page

Instance	Emp	Skill	Act	μ Act	μ TW	PH	T.D.C
C107_100t_20w	20	0.75	100	90.	180.	1236	2,3,1,3,0
C107_25t_5w	5	0.6	25	90.	180.	1236	0,2,0,1,0
C107_50t_10w	10	0.8	50	90.	180.	1236	1,1,1,2,0
C108_100t_20w	20	0.75	100	90.	243.28	1236	4,4,1,3,0
C108_25t_5w	5	0.6	25	90.	242.16	1236	0,2,0,1,0
C108_50t_10w	10	0.8	50	90.	240.78	1236	1,1,1,2,0
C109_100t_20w	20	0.75	100	90.	360.	1236	6,4,1,3,0
C109_25t_5w	5	0.6	25	90.	360.	1236	0,2,0,1,0
C109_50t_10w	10	0.8	50	90.	360.	1236	1,2,1,2,0
C201_100t_20w	20	0.75	100	90.	160.	3390	0,1,1,3,0
C201_25t_5w	5	0.6	25	90.	160.	3390	0,1,0,1,0
C201_50t_10w	10	0.8	50	90.	160.	3390	0,2,1,2,0
C202_100t_20w	20	0.75	100	90.	937.74	3390	6,4,1,3,0
C202_25t_5w	5	0.6	25	90.	1032.28	3390	0,2,0,1,0
C202_50t_10w	10	0.8	50	90.	969.42	3390	0,5,1,2,0
C203_100t_20w	20	0.75	100	90.	1714.82	3390	9,5,1,3,0
C203_25t_5w	5	0.6	25	90.	1778.6	3390	0,3,0,1,0
C203_50t_10w	10	0.8	50	90.	1716.36	3390	1,5,1,2,0
C204_100t_20w	20	0.75	100	90.	2492.58	3390	11,5,1,3,0
C204_25t_5w	5	0.6	25	90.	2277.4	3390	0,3,0,1,0
C204_50t_10w	10	0.8	50	90.	2650.24	3390	1,5,1,2,0
C205_100t_20w	20	0.75	100	90.	320.	3390	5,3,1,3,0
C205_25t_5w	5	0.6	25	90.	320.	3390	0,1,0,1,0
C205_50t_10w	10	0.8	50	90.	320.	3390	0,2,1,2,0
C206_100t_20w	20	0.75	100	90.	486.64	3390	7,3,1,3,0
C206_25t_5w	5	0.6	25	90.	464.52	3390	0,2,0,1,0
C206_50t_10w	10	0.8	50	90.	480.48	3390	0,5,1,2,0
C207_100t_20w	20	0.75	100	90.	612.32	3390	6,3,1,3,0
C207_25t_5w	5	0.6	25	90.	742.	3390	0,2,0,1,0
C207_50t_10w	10	0.8	50	90.	790.8	3390	1,4,1,2,0
C208_100t_20w	20	0.75	100	90.	640.	3390	9,3,1,3,0
C208_25t_5w	5	0.6	25	90.	640.	3390	0,2,0,1,0
C208_50t_10w	10	0.8	50	90.	640.	3390	1,5,1,2,0
R101_100t_20w	20	0.75	100	10.	10.	230	1,0,1,3,0
R101_25t_5w	5	0.6	25	10.	10.	230	0,0,0,1,0
R101_50t_10w	10	0.8	50	10.	10.	230	0,0,1,2,0
R102_100t_20w	20	0.75	100	10.	57.39	230	6,4,1,3,0
R102_25t_5w	5	0.6	25	10.	63.44	230	0,1,0,1,0
R102_50t_10w	10	0.8	50	10.	59.24	230	0,5,1,2,0
R103_100t_20w	20	0.75	100	10.	102.99	230	9,5,1,3,0
R103_25t_5w	5	0.6	25	10.	106.88	230	0,2,0,1,0
R103_50t_10w	10	0.8	50	10.	102.62	230	1,5,1,2,0
R104_100t_20w	20	0.75	100	10.	148.31	230	11,5,1,3,0
R104_25t_5w	5	0.6	25	10.	136.64	230	0,3,0,1,0
R104_50t_10w	10	0.8	50	10.	157.4	230	1,5,1,2,0
R105_100t_20w	20	0.75	100	10.	30.	230	6,0,1,3,0
R105_25t_5w	5	0.6	25	10.	30.	230	0,0,0,1,0
R105_50t_10w	10	0.8	50	10.	30.	230	0,1,1,2,0
R106_100t_20w	20	0.75	100	10.	72.39	230	9,4,1,3,0
R106_25t_5w	5	0.6	25	10.	77.84	230	0,1,0,1,0
R106_50t_10w	10	0.8	50	10.	74.04	230	0,5,1,2,0
R107_100t_20w	20	0.75	100	10.	112.99	230	10,5,1,3,0
R107_25t_5w	5	0.6	25	10.	116.48	230	0,2,0,1,0
R107_50t_10w	10	0.8	50	10.	112.62	230	1,5,1,2,0
R108_100t_20w	20	0.75	100	10.	153.31	230	11,5,1,3,0
R108_25t_5w	5	0.6	25	10.	143.04	230	0,3,0,1,0

Table A.1 – continued from previous page

Instance	Emp	Skill	Act	μ Act	μ TW	PH	T.D.C
R108_50t_10w	10	0.8	50	10.	161.4	230	1,5,1,2,0
R109_100t_20w	20	0.75	100	10.	58.89	230	6,2,1,3,0
R109_25t_5w	5	0.6	25	10.	58.36	230	0,1,0,1,0
R109_50t_10w	10	0.8	50	10.	58.94	230	0,3,1,2,0
R110_100t_20w	20	0.75	100	10.	86.5	230	9,4,1,3,0
R110_25t_5w	5	0.6	25	10.	83.28	230	0,3,0,1,0
R110_50t_10w	10	0.8	50	10.	86.44	230	1,3,1,2,0
R111_100t_20w	20	0.75	100	10.	93.1	230	10,5,1,3,0
R111_25t_5w	5	0.6	25	10.	93.72	230	0,2,0,1,0
R111_50t_10w	10	0.8	50	10.	95.46	230	1,5,1,2,0
R112_100t_20w	20	0.75	100	10.	117.64	230	12,5,1,3,0
R112_25t_5w	5	0.6	25	10.	116.44	230	0,3,0,1,0
R112_50t_10w	10	0.8	50	10.	117.76	230	1,5,1,2,0
R201_100t_20w	20	0.75	100	10.	115.96	1000	2,0,1,3,0
R201_25t_5w	5	0.6	25	10.	113.72	1000	0,0,0,1,0
R201_50t_10w	10	0.8	50	10.	116.46	1000	0,0,1,2,0
R202_100t_20w	20	0.75	100	10.	328.81	1000	7,4,1,3,0
R202_25t_5w	5	0.6	25	10.	352.56	1000	0,1,0,1,0
R202_50t_10w	10	0.8	50	10.	339.96	1000	0,5,1,2,0
R203_100t_20w	20	0.75	100	10.	541.66	1000	9,5,1,3,0
R203_25t_5w	5	0.6	25	10.	554.96	1000	0,2,0,1,0
R203_50t_10w	10	0.8	50	10.	541.54	1000	1,5,1,2,0
R204_100t_20w	20	0.75	100	10.	751.26	1000	11,5,1,3,0
R204_25t_5w	5	0.6	25	10.	694.48	1000	0,3,0,1,0
R204_50t_10w	10	0.8	50	10.	794.32	1000	1,5,1,2,0
R205_100t_20w	20	0.75	100	10.	240.	1000	6,2,1,3,0
R205_25t_5w	5	0.6	25	10.	240.	1000	0,1,0,1,0
R205_50t_10w	10	0.8	50	10.	240.	1000	0,2,1,2,0
R206_100t_20w	20	0.75	100	10.	422.39	1000	9,4,1,3,0
R206_25t_5w	5	0.6	25	10.	444.64	1000	0,1,0,1,0
R206_50t_10w	10	0.8	50	10.	429.64	1000	0,5,1,2,0
R207_100t_20w	20	0.75	100	10.	602.99	1000	10,5,1,3,0
R207_25t_5w	5	0.6	25	10.	617.68	1000	0,2,0,1,0
R207_50t_10w	10	0.8	50	10.	602.62	1000	1,5,1,2,0
R208_100t_20w	20	0.75	100	10.	783.31	1000	11,5,1,3,0
R208_25t_5w	5	0.6	25	10.	733.84	1000	0,3,0,1,0
R208_50t_10w	10	0.8	50	10.	819.4	1000	1,5,1,2,0
R209_100t_20w	20	0.75	100	10.	349.5	1000	8,2,1,3,0
R209_25t_5w	5	0.6	25	10.	332.72	1000	0,3,0,1,0
R209_50t_10w	10	0.8	50	10.	351.08	1000	1,3,1,2,0
R210_100t_20w	20	0.75	100	10.	383.27	1000	9,4,1,3,0
R210_25t_5w	5	0.6	25	10.	385.88	1000	0,1,0,1,0
R210_50t_10w	10	0.8	50	10.	390.06	1000	0,5,1,2,0
R211_100t_20w	20	0.75	100	10.	471.94	1000	12,5,1,3,0
R211_25t_5w	5	0.6	25	10.	467.48	1000	0,3,0,1,0
R211_50t_10w	10	0.8	50	10.	472.92	1000	1,5,1,2,0
RC101_100t_20w	20	0.75	100	10.	30.	240	4,1,1,3,0
RC101_25t_5w	5	0.6	25	10.	30.	240	0,1,0,1,0
RC101_50t_10w	10	0.8	50	10.	30.	240	0,1,1,2,0
RC102_100t_20w	20	0.75	100	10.	71.46	240	8,4,1,3,0
RC102_25t_5w	5	0.6	25	10.	75.4	240	0,1,0,1,0
RC102_50t_10w	10	0.8	50	10.	71.08	240	0,5,1,2,0
RC103_100t_20w	20	0.75	100	10.	112.5	240	10,5,1,3,0
RC103_25t_5w	5	0.6	25	10.	113.92	240	0,2,0,1,0
RC103_50t_10w	10	0.8	50	10.	108.8	240	1,5,1,2,0
RC104_100t_20w	20	0.75	100	10.	154.6	240	11,5,1,3,0

Table A.1 – continued from previous page

Instance	Emp	Skill	Act	μ Act	μ TW	PH	T.D.C
RC104_25t_5w	5	0.6	25	10.	140.24	240	0,3,0,1,0
RC104_50t_10w	10	0.8	50	10.	156.54	240	1,5,1,2,0
RC105_100t_20w	20	0.75	100	10.	54.33	240	7,3,1,3,0
RC105_25t_5w	5	0.6	25	10.	55.28	240	0,1,0,1,0
RC105_50t_10w	10	0.8	50	10.	56.38	240	0,5,1,2,0
RC106_100t_20w	20	0.75	100	10.	60.	240	6,2,1,3,0
RC106_25t_5w	5	0.6	25	10.	60.	240	0,1,0,1,0
RC106_50t_10w	10	0.8	50	10.	60.	240	1,3,1,2,0
RC107_100t_20w	20	0.75	100	10.	88.21	240	9,4,1,3,0
RC107_25t_5w	5	0.6	25	10.	85.96	240	0,3,0,1,0
RC107_50t_10w	10	0.8	50	10.	88.1	240	1,3,1,2,0
RC108_100t_20w	20	0.75	100	10.	112.33	240	12,5,1,3,0
RC108_25t_5w	5	0.6	25	10.	110.48	240	0,3,0,1,0
RC108_50t_10w	10	0.8	50	10.	111.62	240	1,5,1,2,0
RC201_100t_20w	20	0.75	100	10.	120.	960	2,0,1,3,0
RC201_25t_5w	5	0.6	25	10.	120.	960	0,0,0,1,0
RC201_50t_10w	10	0.8	50	10.	120.	960	0,0,1,2,0
RC202_100t_20w	20	0.75	100	10.	318.96	960	6,4,1,3,0
RC202_25t_5w	5	0.6	25	10.	341.8	960	0,1,0,1,0
RC202_50t_10w	10	0.8	50	10.	324.88	960	0,5,1,2,0
RC203_100t_20w	20	0.75	100	10.	517.5	960	9,5,1,3,0
RC203_25t_5w	5	0.6	25	10.	531.52	960	0,2,0,1,0
RC203_50t_10w	10	0.8	50	10.	513.8	960	1,5,1,2,0
RC204_100t_20w	20	0.75	100	10.	717.1	960	11,5,1,3,0
RC204_25t_5w	5	0.6	25	10.	658.64	960	0,3,0,1,0
RC204_50t_10w	10	0.8	50	10.	750.54	960	1,5,1,2,0
RC205_100t_20w	20	0.75	100	10.	223.06	960	5,3,1,3,0
RC205_25t_5w	5	0.6	25	10.	227.76	960	0,1,0,1,0
RC205_50t_10w	10	0.8	50	10.	230.5	960	0,4,1,2,0
RC206_100t_20w	20	0.75	100	10.	240.	960	6,2,1,3,0
RC206_25t_5w	5	0.6	25	10.	240.	960	0,1,0,1,0
RC206_50t_10w	10	0.8	50	10.	240.	960	0,2,1,2,0
RC207_100t_20w	20	0.75	100	10.	349.5	960	9,3,1,3,0
RC207_25t_5w	5	0.6	25	10.	332.72	960	0,3,0,1,0
RC207_50t_10w	10	0.8	50	10.	351.08	960	1,3,1,2,0
RC208_100t_20w	20	0.75	100	10.	471.93	960	12,5,1,3,0
RC208_25t_5w	5	0.6	25	10.	467.44	960	0,3,0,1,0
RC208_50t_10w	10	0.8	50	10.	472.9	960	1,5,1,2,0
test150-0-0-0-0_d0_tw0	103	0.76	150	20.47	480.	480	11,15,4,4,3
test150-0-0-0-0_d0_tw1	103	0.76	150	20.47	160.	480	3,3,4,4,3
test150-0-0-0-0_d0_tw2	103	0.76	150	20.47	127.27	480	3,3,4,4,3
test150-0-0-0-0_d0_tw3	103	0.76	150	20.47	100.	480	3,3,4,4,3
test150-0-0-0-0_d0_tw4	103	0.76	150	20.47	155.93	480	5,9,4,4,3
test250-0-0-0-0_d0_tw0	171	0.68	250	20.44	480.	480	18,17,7,4,7
test250-0-0-0-0_d0_tw1	171	0.68	250	20.44	160.	480	6,5,7,4,7
test250-0-0-0-0_d0_tw2	171	0.68	250	20.44	127.12	480	6,5,7,4,7
test250-0-0-0-0_d0_tw3	171	0.68	250	20.44	100.	480	6,5,7,4,7
test250-0-0-0-0_d0_tw4	171	0.68	250	20.44	154.4	480	4,6,7,4,7
test50-0-0-0-0_d0_tw0	38	0.74	50	22.6	480.	480	5,5,1,2,0
test50-0-0-0-0_d0_tw1	38	0.74	50	22.6	160.	480	3,1,1,2,0
test50-0-0-0-0_d0_tw2	38	0.74	50	22.6	128.4	480	3,1,1,2,0
test50-0-0-0-0_d0_tw3	38	0.74	50	22.6	100.	480	3,1,1,2,0
test50-0-0-0-0_d0_tw4	38	0.74	50	22.6	152.	480	2,4,1,2,0
hh_00_P0	15	0.94	153	31.96	106.41	1380	2,1,0,0,0
ll1_00_P0	9	0.99	106	24.62	65.67	1380	0,0,0,0,0
ll1_01_P0	9	0.99	106	24.62	65.67	1380	1,0,0,0,0

Table A.1 – continued from previous page

Instance	Emp	Skill	Act	μ Act	μ TW	PH	T.D.C
ll1_02_P0	9	0.99	106	24.62	65.67	1380	0,0,0,0,0
ll1_03_P0	9	0.99	106	24.62	65.67	1380	0,0,0,0,0
ll1_04_P0	9	0.99	106	24.62	65.67	1380	0,0,0,0,0
ll1_05_P0	9	0.99	106	24.62	65.67	1380	0,1,0,0,0
ll1_06_P0	9	0.99	106	24.62	65.67	1380	7,0,0,0,0
ll1_07_P0	9	0.99	106	24.62	65.67	1380	0,0,0,0,0
ll2_00_P0	7	0.99	60	30.03	58.33	1380	0,0,0,0,0
ll3_00_P0	7	0.99	60	30.05	58.23	1380	0,0,0,0,0

Appendix B

Result of experiments - IP Model

B.1 Results with teaming & time-dependencies

Table B.1

Instance Name	Time	Best Obj	Bound	Gap	Instance Name	Time	Best Obj	Bound	Gap
C101_25t_5w	0.62	116.08	116.08	0	R112_25t_5w	900.01	-	156.3627	-
C101_50t_10w	252.33	242.34	242.3304	0	R112_50t_10w	900.89	-	224.0162	-
C102_25t_5w	900.02	113.904	88.686	0.22	R201_25t_5w	900.03	256.404	241.611	0.06
C102_50t_10w	900.06	-	224.6443	-	R201_50t_10w	900.02	-	380.9798	-
C103_25t_5w	900.02	-	74.2218	-	R202_25t_5w	900.01	280.772	187.3785	0.33
C103_50t_10w	900.06	-	145.4992	-	R202_50t_10w	900.02	-	299.711	-
C104_25t_5w	900.11	-	70.735	-	R203_25t_5w	900	-	174.2657	-
C104_50t_10w	901.21	-	137.4903	-	R203_50t_10w	900.35	-	255.5945	-
C105_25t_5w	3.86	124.06	124.06	0	R204_25t_5w	900.42	-	163.1976	-
C105_50t_10w	522.08	265.756	265.756	0	R204_50t_10w	901.2	-	224.6658	-
C106_25t_5w	0.67	116.08	116.08	0	R205_25t_5w	900	224.936	188.3601	0.16
C106_50t_10w	588.78	242.34	242.3186	0	R205_50t_10w	900.03	-	292.4445	-
C107_25t_5w	2.78	124.06	124.06	0	R206_25t_5w	900	-	164.4554	-
C107_50t_10w	900.03	257.12	249.6667	0.03	R206_50t_10w	900.24	-	268.4518	-
C108_25t_5w	900.01	124.516	99.1684	0.2	R207_25t_5w	900.51	-	159.7039	-
C108_50t_10w	900.13	-	164.3347	-	R207_50t_10w	900.1	-	244.6857	-
C109_25t_5w	900.78	-	55.8916	-	R208_25t_5w	900.01	-	159.9671	-
C109_50t_10w	900.05	-	107.4942	-	R208_50t_10w	900.95	-	224.2771	-
C201_25t_5w	1.63	146.804	146.804	0	R209_25t_5w	900.01	400.716	187.5394	0.53
C201_50t_10w	44.99	257.016	257.016	0	R209_50t_10w	905.92	-	262.0608	-
C202_25t_5w	900.01	149.852	110.8134	0.26	R210_25t_5w	900.03	-	187.9183	-
C202_50t_10w	900.81	-	204.4078	-	R210_50t_10w	900.06	-	264.6105	-
C203_25t_5w	900.02	209.612	105.3453	0.5	R211_25t_5w	900.32	-	156.7452	-
C203_50t_10w	900.35	-	176.8458	-	R211_50t_10w	900.16	-	223.196	-
C204_25t_5w	900	-	90.5569	-	RC101_25t_5w	900.01	-	228.76	-
C204_50t_10w	902.6	-	154.8397	-	RC101_50t_10w	901.31	-	377.8716	-
C205_25t_5w	900	148.704	116.3554	0.22	RC102_25t_5w	900.02	-	178.4969	-
C205_50t_10w	910.6	-	207.1381	-	RC102_50t_10w	188.63	-	-	Infeasible
C206_25t_5w	900.04	151.084	110.0439	0.27	RC103_25t_5w	900.03	-	91.9598	-
C206_50t_10w	900.03	-	200.1382	-	RC103_50t_10w	900.15	-	132.8921	-
C207_25t_5w	900.03	204.98	121.703	0.41	RC104_25t_5w	900.01	-	71.0499	-
C207_50t_10w	900.39	-	213.7431	-	RC104_50t_10w	900.29	-	108.9939	-
C208_25t_5w	900.05	195.132	96.6801	0.5	RC105_25t_5w	900.01	-	179.5345	-
C208_50t_10w	900.83	-	192.8239	-	RC105_50t_10w	900.07	-	283.1036	-
R101_25t_5w	1.53	-	-	Infeasible	RC106_25t_5w	900.01	-	124.8901	-
R101_50t_10w	20.05	-	-	Infeasible	RC106_50t_10w	900.02	-	174.9441	-
R102_25t_5w	900	-	241.2626	-	RC107_25t_5w	900.01	-	67.07	-
R102_50t_10w	82.75	-	-	Infeasible	RC107_50t_10w	901.06	-	108.7505	-
R103_25t_5w	900.19	-	205.7702	-	RC108_25t_5w	900.03	-	60.2942	-
R103_50t_10w	104.99	-	-	Infeasible	RC108_50t_10w	900.07	-	104.2573	-
R104_25t_5w	900.01	-	187.6477	-	RC201_25t_5w	900.01	230.808	209.3653	0.09
R104_50t_10w	296.3	-	-	Infeasible	RC201_50t_10w	900.15	-	298.7893	-
R105_25t_5w	4.78	-	-	Infeasible	RC202_25t_5w	900.01	230.112	154.4222	0.33
R105_50t_10w	900.08	-	450.7153	-	RC202_50t_10w	900.45	-	188.7285	-
R106_25t_5w	900	-	212.0484	-	RC203_25t_5w	900	-	79.1323	-
R106_50t_10w	900.02	-	335.8864	-	RC203_50t_10w	900.07	-	128.4802	-
R107_25t_5w	900.06	-	194.0732	-	RC204_25t_5w	900.01	-	65.323	-
R107_50t_10w	900.04	-	262.2022	-	RC204_50t_10w	901.61	-	108.9149	-
R108_25t_5w	900	-	185.1993	-	RC205_25t_5w	900.04	223.84	176.3638	0.21
R108_50t_10w	902.24	-	233.6151	-	RC205_50t_10w	900.15	-	239.5656	-
R109_25t_5w	900.01	-	213.1001	-	RC206_25t_5w	900.07	214.368	88.2789	0.59
R109_50t_10w	900.16	-	290.8524	-	RC206_50t_10w	900.25	-	149.3614	-
R110_25t_5w	900	-	181.3772	-	RC207_25t_5w	900.01	-	75.4133	-
R110_50t_10w	900.97	-	243.6301	-	RC207_50t_10w	900.14	-	124.8226	-
R111_25t_5w	902.19	-	202.6721	-	RC208_25t_5w	900.01	-	59.4431	-
R111_50t_10w	901.19	-	260.3502	-	RC208_50t_10w	900.09	-	103.9515	-

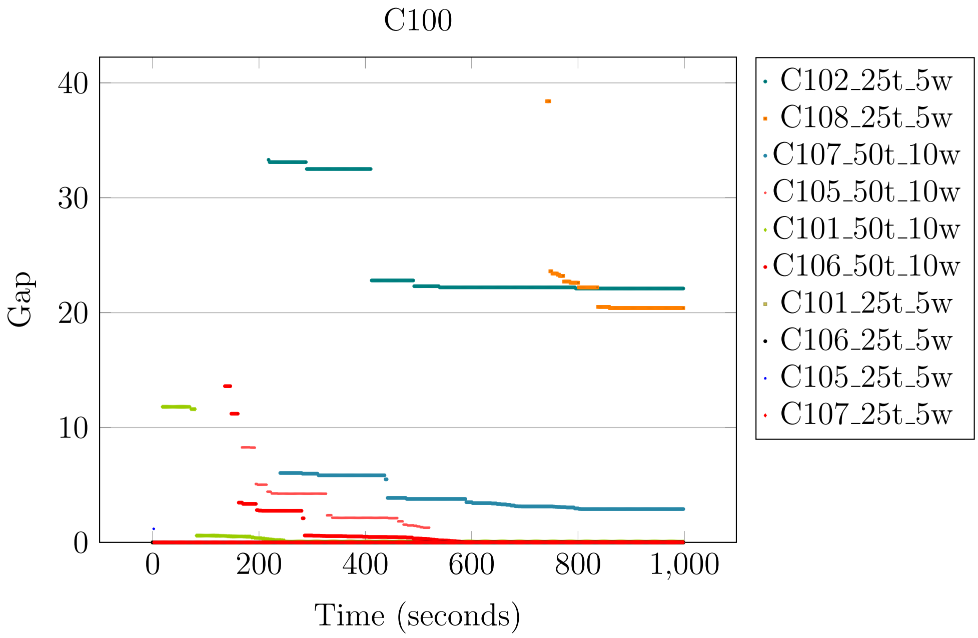

Figure B.1: Cl00 with teaming and time-dependent constraints (15 min time limit).

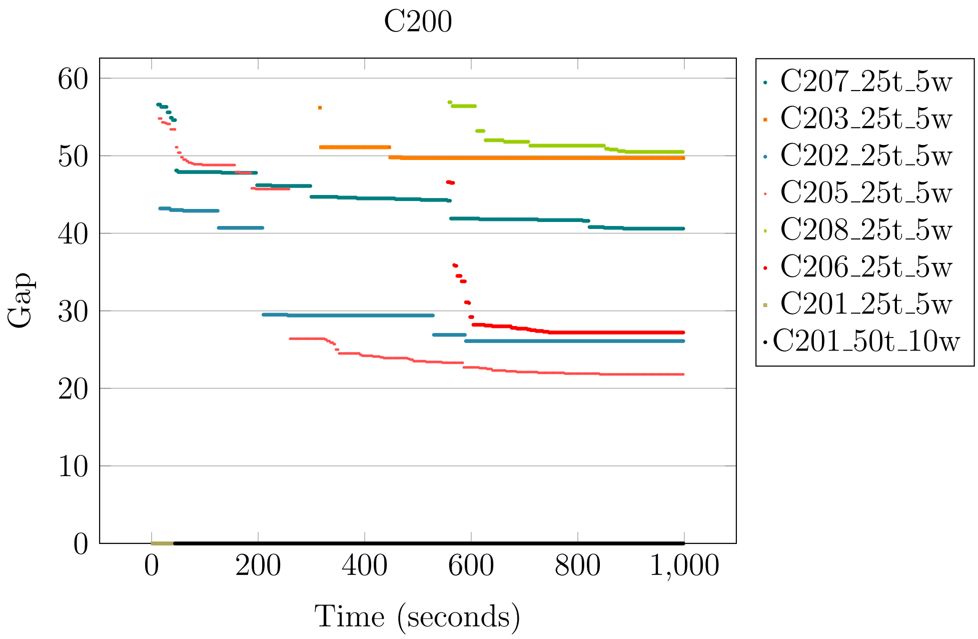

Figure B.2: C200 with teaming and time-dependent constraints (15 min time limit).

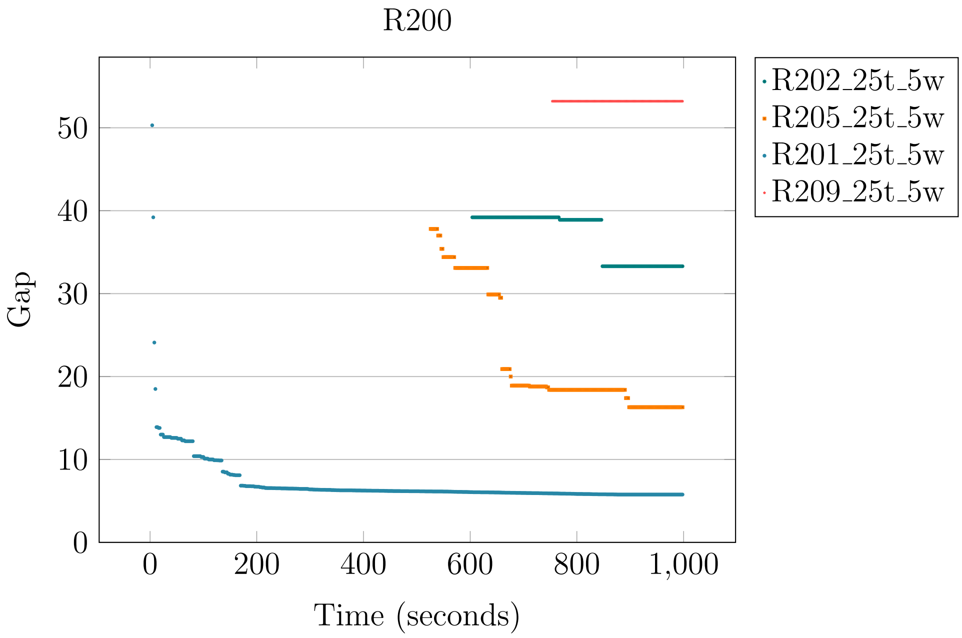

Figure B.3: R200 with teaming and time-dependent constraints (15 min time limit).

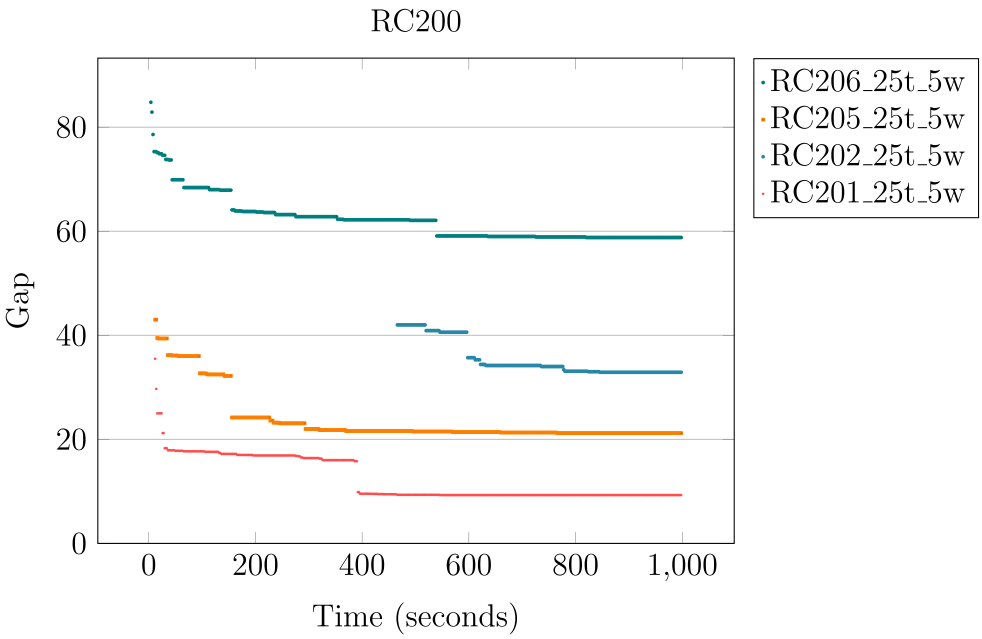

Figure B.4: RC200 with teaming and time-dependent constraints (15 min time limit).

B.2 Results without teaming & time-dependencies

Table B.2

Instance Name	Time	Best Obj	Bound	Gap	Instance Name	Time	Best Obj	Bound	Gap
C101_25t_5w	0.34	8.39E+01	8.39E+01	0	R112_25t_5w	900.01	-	1.26E+02	-
C101_50t_10w	15.02	1.60E+02	1.60E+02	0	R112_50t_10w	900.07	-	1.90E+02	-
C102_25t_5w	900.16	8.34E+01	7.31E+01	0.12	R201_25t_5w	4.53	1.93E+02	1.93E+02	0
C102_50t_10w	900.02	-	1.47E+02	-	R201_50t_10w	900.04	3.34E+02	3.21E+02	0.04
C103_25t_5w	900.01	1.06E+02	6.31E+01	0.4	R202_25t_5w	900.00	1.78E+02	1.49E+02	0.16
C103_50t_10w	900.04	-	1.23E+02	-	R202_50t_10w	900.06	-	2.34E+02	-
C104_25t_5w	900.00	8.55E+01	6.09E+01	0.29	R203_25t_5w	900.03	2.26E+02	1.36E+02	0.4
C104_50t_10w	900.08	-	9.85E+01	-	R203_50t_10w	900.50	-	2.04E+02	-
C105_25t_5w	0.73	8.39E+01	8.39E+01	0	R204_25t_5w	900.32	1.56E+02	1.28E+02	0.18
C105_50t_10w	15.20	1.59E+02	1.59E+02	0	R204_50t_10w	900.59	-	1.92E+02	-
C106_25t_5w	0.39	8.39E+01	8.39E+01	0	R205_25t_5w	900.01	1.66E+02	1.55E+02	0.07
C106_50t_10w	23.23	1.60E+02	1.60E+02	0	R205_50t_10w	900.06	-	2.52E+02	-
C107_25t_5w	1.06	8.39E+01	8.39E+01	0	R206_25t_5w	900.01	1.74E+02	1.39E+02	0.2
C107_50t_10w	18.78	1.59E+02	1.59E+02	0	R206_50t_10w	901.70	-	2.17E+02	-
C108_25t_5w	900.00	8.39E+01	8.00E+01	0.05	R207_25t_5w	900.07	1.82E+02	1.32E+02	0.28
C108_50t_10w	911.88	-	1.25E+02	-	R207_50t_10w	900.29	-	1.98E+02	-
C109_25t_5w	900.00	8.59E+01	4.99E+01	0.42	R208_25t_5w	900.03	1.58E+02	1.28E+02	0.19
C109_50t_10w	900.10	-	8.42E+01	-	R208_50t_10w	900.32	-	1.90E+02	-
C201_25t_5w	0.45	9.35E+01	9.35E+01	0	R209_25t_5w	900.03	1.62E+02	1.39E+02	0.14
C201_50t_10w	11.53	1.60E+02	1.60E+02	0	R209_50t_10w	900.07	-	2.22E+02	-
C202_25t_5w	330.68	9.35E+01	9.35E+01	0	R210_25t_5w	900.03	1.82E+02	1.49E+02	0.18
C202_50t_10w	900.03	1.64E+02	1.45E+02	0.12	R210_50t_10w	900.07	-	2.18E+02	-
C203_25t_5w	900.03	1.06E+02	7.93E+01	0.25	R211_25t_5w	900.02	1.51E+02	1.27E+02	0.16
C203_50t_10w	900.10	-	1.37E+02	-	R211_50t_10w	900.05	-	1.90E+02	-
C204_25t_5w	900.24	1.22E+02	7.18E+01	0.41	RC101_25t_5w	466.83	1.94E+02	1.94E+02	0
C204_50t_10w	900.11	2.55E+02	1.25E+02	0.51	RC101_50t_10w	900.03	-	3.16E+02	-
C205_25t_5w	16.86	9.35E+01	9.35E+01	0	RC102_25t_5w	900.01	1.51E+02	1.22E+02	0.2
C205_50t_10w	900.75	4.27E+02	1.52E+02	0.64	RC102_50t_10w	900.04	-	1.81E+02	-
C206_25t_5w	900.00	9.35E+01	9.21E+01	0.02	RC103_25t_5w	900.01	1.42E+02	5.60E+01	0.6
C206_50t_10w	900.25	2.45E+02	1.40E+02	0.43	RC103_50t_10w	900.93	-	1.10E+02	-
C207_25t_5w	900.02	9.68E+01	8.53E+01	0.12	RC104_25t_5w	900.01	1.33E+02	5.26E+01	0.61
C207_50t_10w	900.47	-	1.49E+02	-	RC104_50t_10w	900.82	-	9.52E+01	-
C208_25t_5w	900.02	9.42E+01	7.87E+01	0.16	RC105_25t_5w	900.01	1.72E+02	1.31E+02	0.24
C208_50t_10w	900.02	-	1.39E+02	-	RC105_50t_10w	900.01	-	2.07E+02	-
R101_25t_5w	0.42	-	-	Infeasible	RC106_25t_5w	900.01	1.51E+02	1.09E+02	0.28
R101_50t_10w	9.94	-	-	Infeasible	RC106_50t_10w	900.06	-	1.39E+02	-
R102_25t_5w	900.00	-	1.95E+02	-	RC107_25t_5w	900.18	1.28E+02	5.12E+01	0.6
R102_50t_10w	900.10	-	3.02E+02	-	RC107_50t_10w	900.06	-	9.55E+01	-
R103_25t_5w	900.00	-	1.53E+02	-	RC108_25t_5w	900.06	1.30E+02	4.82E+01	0.63
R103_50t_10w	900.05	-	2.33E+02	-	RC108_50t_10w	900.98	-	9.17E+01	-
R104_25t_5w	900.00	-	1.38E+02	-	RC201_25t_5w	33.88	1.51E+02	1.51E+02	0
R104_50t_10w	900.24	-	2.01E+02	-	RC201_50t_10w	900.03	-	2.50E+02	-
R105_25t_5w	900.00	-	2.26E+02	-	RC202_25t_5w	900.01	1.50E+02	1.03E+02	0.32
R105_50t_10w	902.52	3.90E+02	3.69E+02	0.05	RC202_50t_10w	900.09	-	1.37E+02	-
R106_25t_5w	900.00	1.98E+02	1.71E+02	0.13	RC203_25t_5w	900.03	1.64E+02	5.57E+01	0.66
R106_50t_10w	900.22	-	2.47E+02	-	RC203_50t_10w	900.07	-	1.09E+02	-
R107_25t_5w	900.00	-	1.44E+02	-	RC204_25t_5w	900.26	1.32E+02	5.02E+01	0.62
R107_50t_10w	900.10	-	2.13E+02	-	RC204_50t_10w	902.07	-	9.50E+01	-
R108_25t_5w	900.00	-	1.38E+02	-	RC205_25t_5w	900.01	1.45E+02	1.16E+02	0.2
R108_50t_10w	900.86	-	1.98E+02	-	RC205_50t_10w	902.25	3.19E+02	1.79E+02	0.44
R109_25t_5w	900.00	1.84E+02	1.76E+02	0.04	RC206_25t_5w	900.01	1.38E+02	7.62E+01	0.45
R109_50t_10w	900.01	-	2.55E+02	-	RC206_50t_10w	900.03	-	1.36E+02	-
R110_25t_5w	900.00	-	1.38E+02	-	RC207_25t_5w	900.01	1.29E+02	6.43E+01	0.5
R110_50t_10w	901.85	-	2.09E+02	-	RC207_50t_10w	900.03	8.89E+02	1.08E+02	0.88
R111_25t_5w	900.02	-	1.50E+02	-	RC208_25t_5w	900.07	1.25E+02	4.78E+01	0.62
R111_50t_10w	900.33	-	2.09E+02	-	RC208_50t_10w	900.76	-	9.29E+01	-

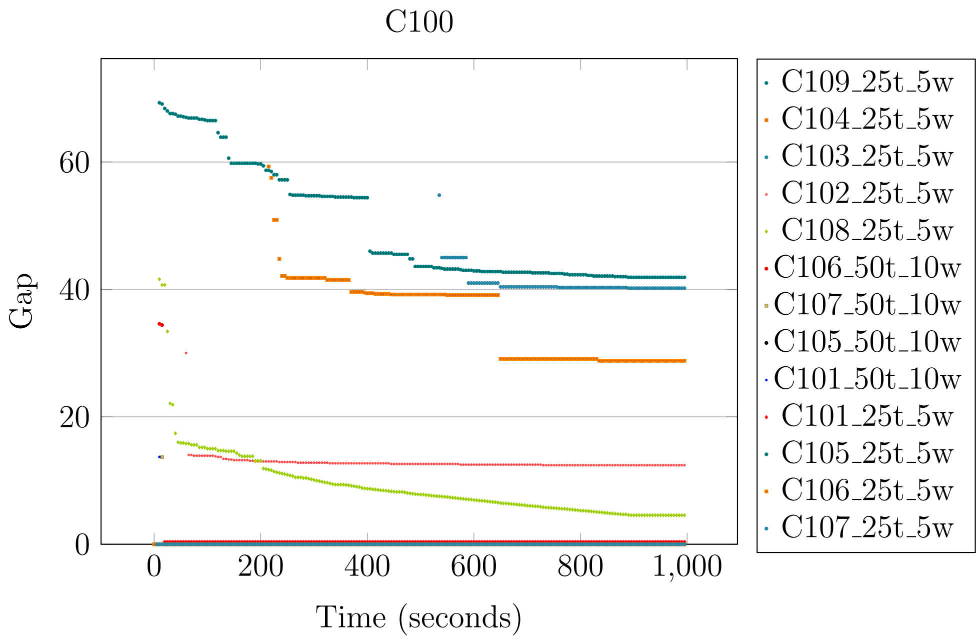

Figure B.5: Cl00 without Teaming and Connected Activities constraints and 15 min time limit.

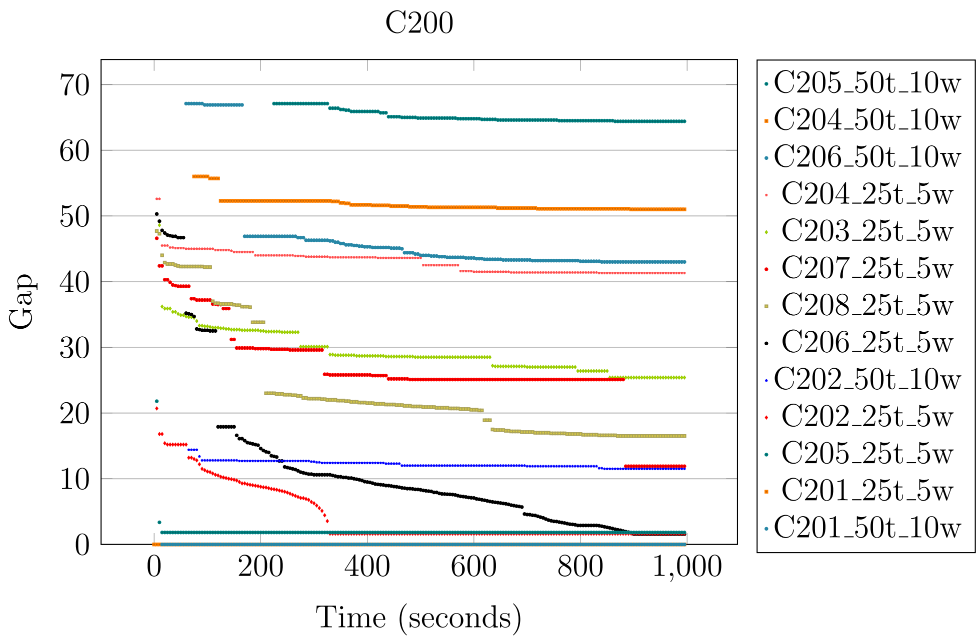

Figure B.6: C200 without Teaming and Connected Activities constraints and 15 min time limit.

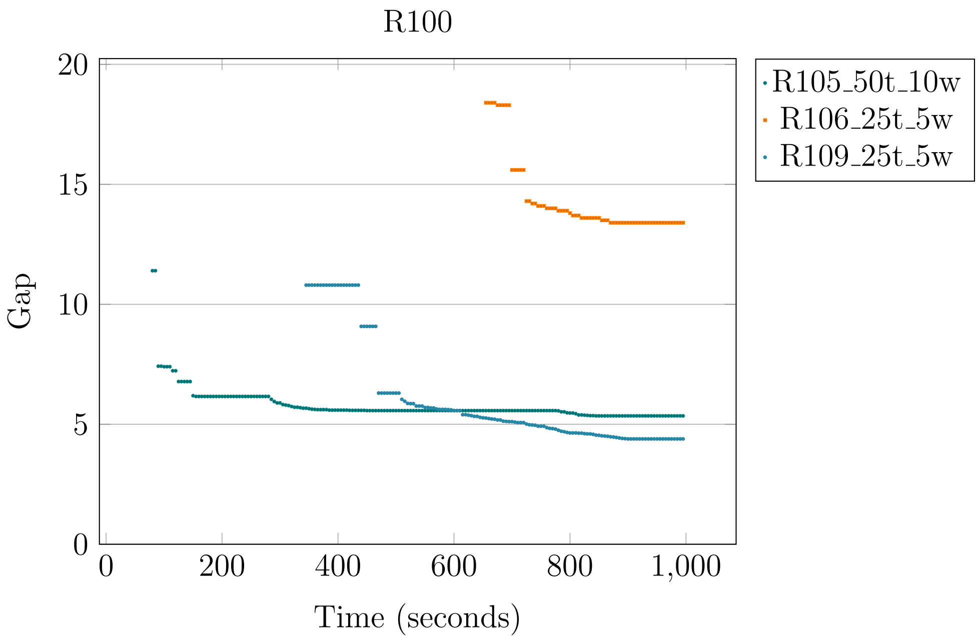

Figure B.7: R100 without Teaming and Connected Activities constraints and 15 min time limit.

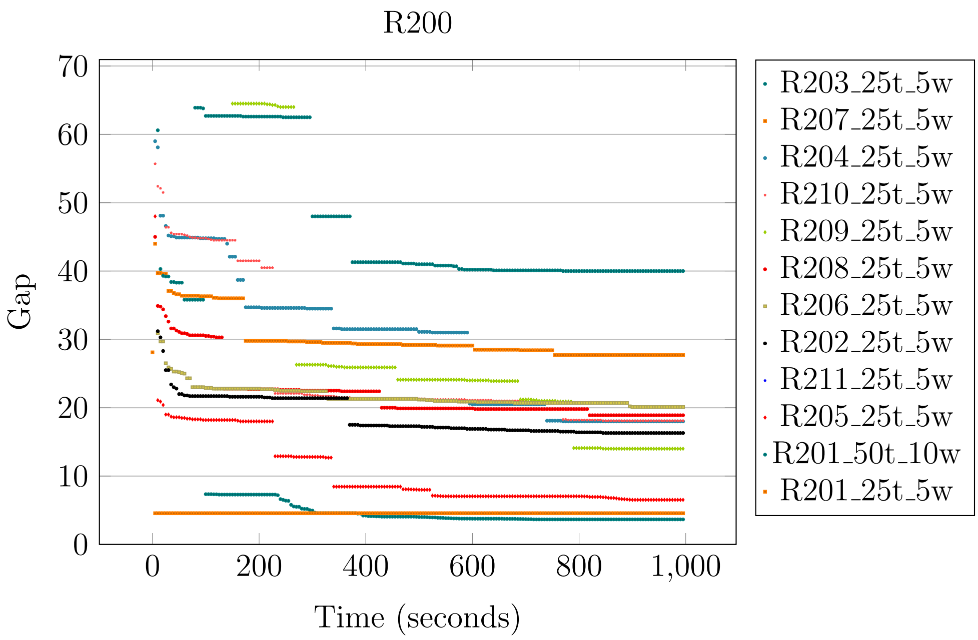

Figure B.8: R200 without Teaming and Connected Activities constraints and 15 min time limit.

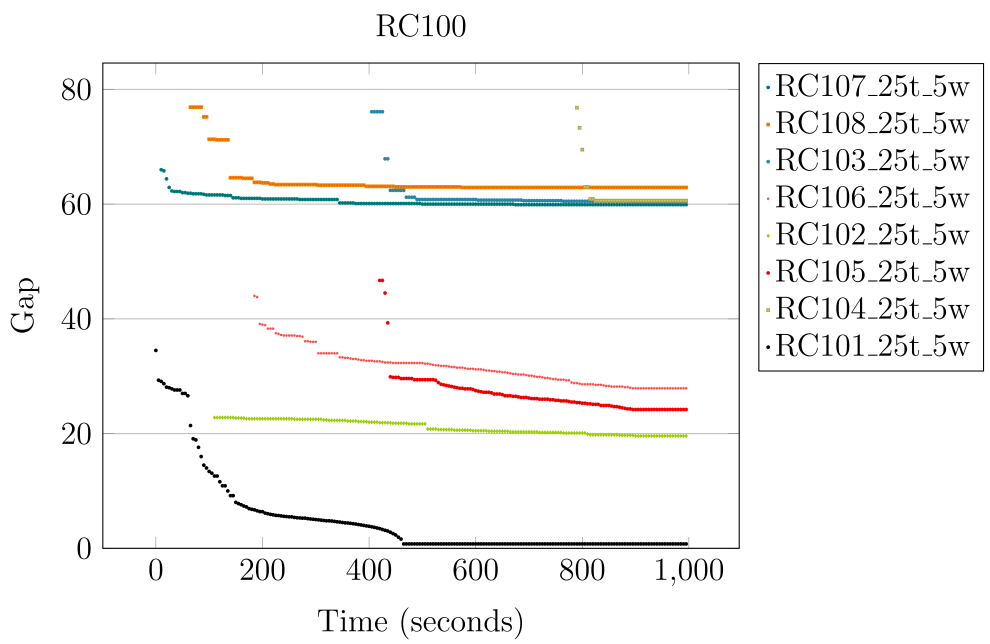

Figure B.9: RCl00 without Teaming and Connected Activities constraints and 15 min time limit.

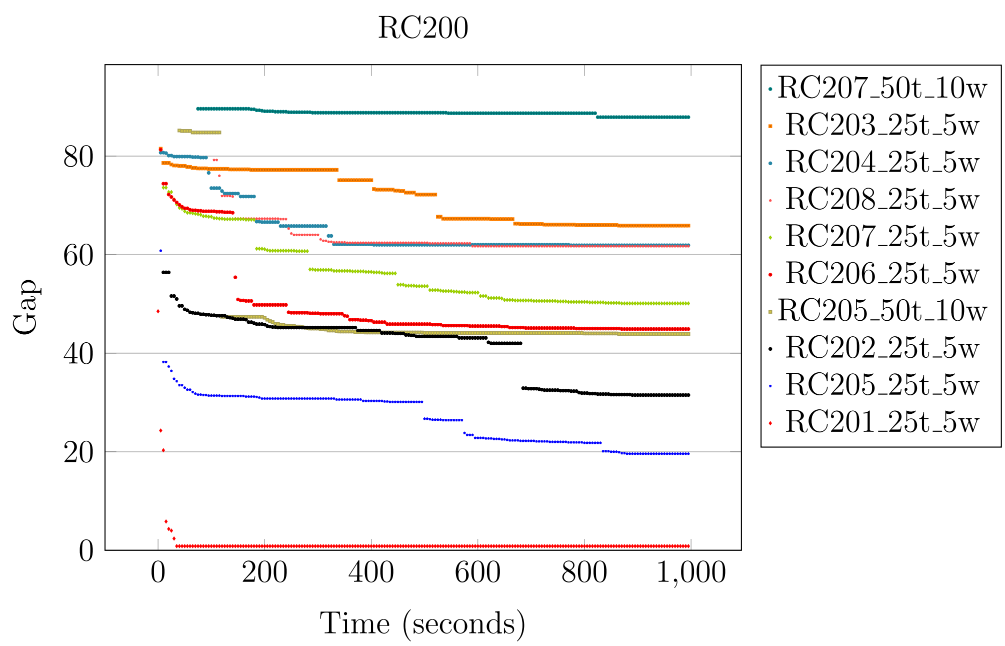

Figure B.10: RC200 without Teaming and Connected Activities constraints and 15 min time limit.

B.3 Teaming & time-dependencies (limit 1 hour)

Instance Name	Time	Best Obj	Bound	Gap	Instance Name	Time	Best Obj	Bound	Gap
C102_50t_10w	3600.04	-	2.25E+02	-	R205_50t_10w	3600.03	-	2.99E+02	-
C103_25t_5w	3600.01	-	7.49E+01	-	R206_25t_5w	3600.01	-	1.68E+02	-
C103_50t_10w	3600.07	-	1.71E+02	-	R206_50t_10w	3600.08	-	2.71E+02	-
C104_25t_5w	3600.79	-	8.81E+01	-	R207_25t_5w	3600.01	-	1.61E+02	-
C104_50t_10w	3600.04	-	1.39E+02	-	R207_50t_10w	3600.16	-	2.45E+02	-
C108_50t_10w	3604.43	-	1.68E+02	-	R208_25t_5w	3600.03	-	1.61E+02	-
C109_25t_5w	3600.02	-	6.15E+01	-	R208_50t_10w	3600.03	-	2.25E+02	-
C109_50t_10w	3600.05	-	1.08E+02	-	R209_50t_10w	3600.09	-	2.63E+02	-
C202_50t_10w	3600.01	-	2.11E+02	-	R210_25t_5w	3600.05	2.76E+02	1.89E+02	0.32
C203_50t_10w	3600.08	-	1.79E+02	-	R210_50t_10w	3600.05	-	2.75E+02	-
C204_25t_5w	3600.04	-	9.12E+01	-	R211_25t_5w	3600.01	-	1.57E+02	-
C204_50t_10w	3600.35	-	1.56E+02	-	R211_50t_10w	3600.03	-	2.26E+02	-
C205_50t_10w	3600.02	-	2.08E+02	-	RC101_25t_5w	3600	-	2.34E+02	-
C206_50t_10w	3600.02	-	2.00E+02	-	RC101_50t_10w	3600.03	-	3.82E+02	-
C207_50t_10w	3600.02	-	2.14E+02	-	RC102_25t_5w	3600.01	-	1.80E+02	-
C208_50t_10w	3600.05	-	1.93E+02	-	RC103_25t_5w	3600.01	-	9.53E+01	-
R102_25t_5w	3600.02	-	2.44E+02	-	RC103_50t_10w	3600.06	-	1.34E+02	-
R103_25t_5w	3600.01	-	2.10E+02	-	RC104_25t_5w	3600.01	-	7.25E+01	-
R104_25t_5w	3600.3	-	1.88E+02	-	RC104_50t_10w	3602.23	-	1.10E+02	-
R105_50t_10w	3600.07	-	4.56E+02	-	RC105_25t_5w	3600.01	-	1.81E+02	-
R106_25t_5w	3600.01	-	2.14E+02	-	RC105_50t_10w	3600.03	-	2.83E+02	-
R106_50t_10w	3600.33	-	3.41E+02	-	RC106_25t_5w	3600.03	2.16E+02	1.32E+02	0.39
R107_25t_5w	3600.01	-	1.97E+02	-	RC106_50t_10w	3600.03	-	1.78E+02	-
R107_50t_10w	3600.08	-	2.68E+02	-	RC107_25t_5w	3600.01	-	6.80E+01	-
R108_25t_5w	3600.01	-	1.87E+02	-	RC107_50t_10w	3600.06	-	1.10E+02	-
R108_50t_10w	3600.82	-	2.38E+02	-	RC108_25t_5w	3600.03	-	6.07E+01	-
R109_25t_5w	3600.01	-	2.18E+02	-	RC108_50t_10w	3600.08	-	1.05E+02	-
R109_50t_10w	3607.16	-	2.97E+02	-	RC201_50t_10w	3600.06	-	3.00E+02	-
R110_25t_5w	3600.02	-	1.83E+02	-	RC202_50t_10w	3600.84	-	2.07E+02	-
R110_50t_10w	3600.02	-	2.46E+02	-	RC203_25t_5w	3600.01	-	8.24E+01	-
R111_25t_5w	3600.01	-	2.05E+02	-	RC203_50t_10w	3600.03	-	1.30E+02	-
R111_50t_10w	3600.15	-	2.69E+02	-	RC204_25t_5w	3600.01	-	6.61E+01	-
R112_25t_5w	3600.22	-	1.57E+02	-	RC204_50t_10w	3600.05	-	1.09E+02	-
R112_50t_10w	3600.05	-	2.26E+02	-	RC205_50t_10w	3600.06	-	2.41E+02	-
R201_50t_10w	3600.05	4.35E+02	3.82E+02	0.12	RC206_50t_10w	3600.03	-	1.50E+02	-
R202_50t_10w	3600.09	-	3.10E+02	-	RC207_25t_5w	3600.01	-	7.71E+01	-
R203_25t_5w	3600	-	1.75E+02	-	RC207_50t_10w	3600.06	-	1.25E+02	-
R203_50t_10w	3600.16	-	2.58E+02	-	RC208_25t_5w	3600.03	-	5.99E+01	-
R204_25t_5w	3600.03	-	1.65E+02	-	RC208_50t_10w	3600.03	-	1.05E+02	-
R204_50t_10w	3600.36	-	2.26E+02	-					

Table B.3

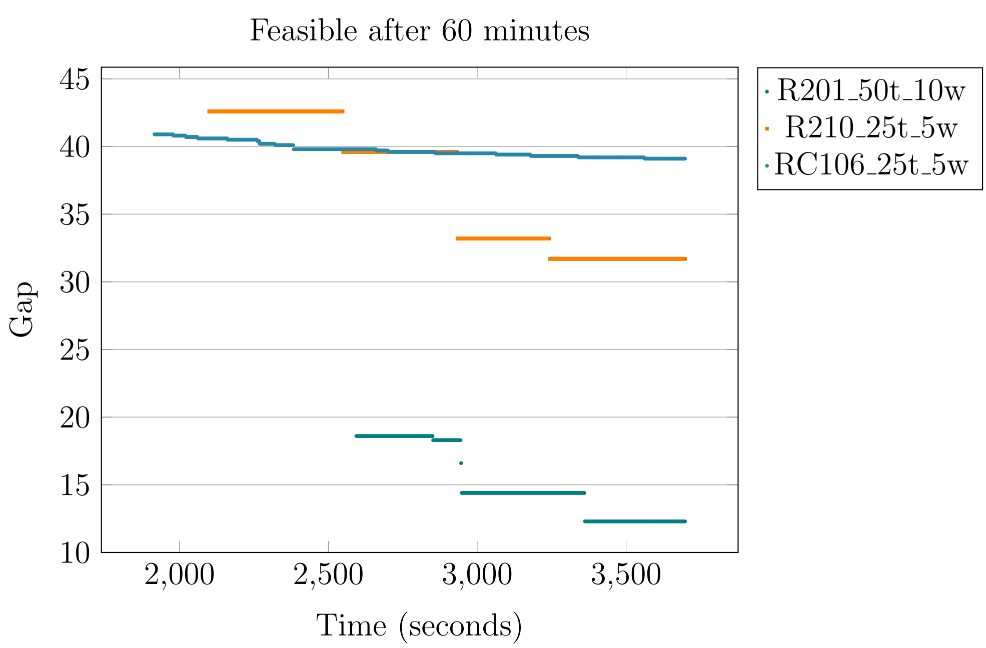

Figure B.11: Solver solutions after increasing time limit to 60 min.

B.4 Teaming & time-dependencies (limit 4 hours)

Instance Name	Time	Best Obj	Bound	Gap	Instance Name	Time	Best Obj	Bound	Gap
C102_50t_10w	14400	265.824	225.08593	0.153252	R204_50t_10w	14400.3	-	225.90787	-
C103_25t_5w	14400	135.776	78.15259	0.424401	R205_50t_10w	14400.4	-	299.96596	-
C103_50t_10w	14400	-	172.61957	-	R206_25t_5w	14400.0	224.388	170.86254	0.23854
C104_25t_5w	14400	130.216	93.56153	0.28149	R206_50t_10w	14400.0	-	274.46083	-
C104_50t_10w	14400	-	140.10743	-	R207_25t_5w	14400.0	-	162.42655	-
C108_50t_10w	14400	-	177.63550	-	R207_50t_10w	14400.9	-	247.52441	-
C109_25t_5w	14400	126.332	65.52286	0.481344	R208_25t_5w	14400.0	232.492	162.50627	0.30102
C109_50t_10w	14400	-	116.68118	-	R208_50t_10w	14400.0	-	224.92155	-
C202_50t_10w	14400	-	216.02025	-	R209_50t_10w	14400.4	-	264.51072	-
C203_50t_10w	14400	-	179.24373	-	R210_50t_10w	14400.0	-	275.54034	-
C204_25t_5w	14400	159.928	94.04139	0.411977	R211_25t_5w	14400.0	247.388	158.45162	0.35950
C204_50t_10w	14400	-	155.08782	-	R211_50t_10w	14400.0	-	226.77157	-
C205_50t_10w	14400	-	202.41043	-	RC101_25t_5w	14400.0	-	258.51817	-
C206_50t_10w	14400	284.304	200.26899	0.295581	RC101_50t_10w	14400.0	458.104	392.72032	0.14272
C207_50t_10w	14400	-	213.59567	-	RC102_25t_5w	14400.0	-	183.29947	-
C208_50t_10w	14400	-	194.146719	-	RC103_25t_5w	14400.0	-	100.61711	-
R102_25t_5w	14400	-	249.16771	-	RC103_50t_10w	14400.0	-	141.47467	-
R103_25t_5w	14400	-	215.35747	-	RC104_25t_5w	14400.0	-	76.121035	-
R104_25t_5w	14400	-	194.16938	-	RC104_50t_10w	14400.0	-	110.83161	-
R105_50t_10w	14400	-	455.54133	-	RC105_25t_5w	14400.0	-	189.92938	-
R106_25t_5w	14400	-	217.68254	-	RC105_50t_10w	14400.0	-	285.30623	-
R106_50t_10w	14400	-	342.20779	-	RC106_50t_10w	14400.0	-	179.28746	-
R107_25t_5w	14400	-	203.38693	-	RC107_25t_5w	14400.0	-	70.43331	-
R107_50t_10w	14400	-	272.33443	-	RC107_50t_10w	14400.1	-	110.14980	-
R108_25t_5w	14400	-	189.48152	-	RC108_25t_5w	14400.0	-	62.200693	-
R108_50t_10w	14400	-	240.17279	-	RC108_50t_10w	14400.0	-	105.68588	-
R109_25t_5w	14400	-	226.79629	-	RC201_50t_10w	14400.0	432.864	304.87446	0.29568
R109_50t_10w	14400	-	299.60077	-	RC202_50t_10w	14400.0	-	215.93072	-
R110_25t_5w	14400	-	186.79719	-	RC203_25t_5w	14400.0	227.544	88.411068	0.61145
R110_50t_10w	14400	-	246.37104	-	RC203_50t_10w	14400.0	-	130.09251	-
R111_25t_5w	14400	-	206.04275	-	RC204_25t_5w	14400.0	220.724	71.381029	0.67660
R111_50t_10w	14400	-	268.80196	-	RC204_50t_10w	14400.0	-	109.56101	-
R112_25t_5w	14400	-	158.89112	-	RC205_50t_10w	14400.0	-	241.98986	-
R112_50t_10w	14400	-	227.26095	-	RC206_50t_10w	14400.0	-	150.59880	-
R202_50t_10w	14400	-	310.45724	-	RC207_25t_5w	14400.0	217.608	86.860989	0.60083
R203_25t_5w	14400	238.664	177.19099	0.257571	RC207_50t_10w	14400.0	-	127.76435	-
R203_50t_10w	14400	-	258.83125	-	RC208_25t_5w	14400.0	-	60.842811	-
R204_25t_5w	14400	241.28	168.15897	0.303055	RC208_50t_10w	14400.0	-	105.68961	-

Table B.4: 240 minutes

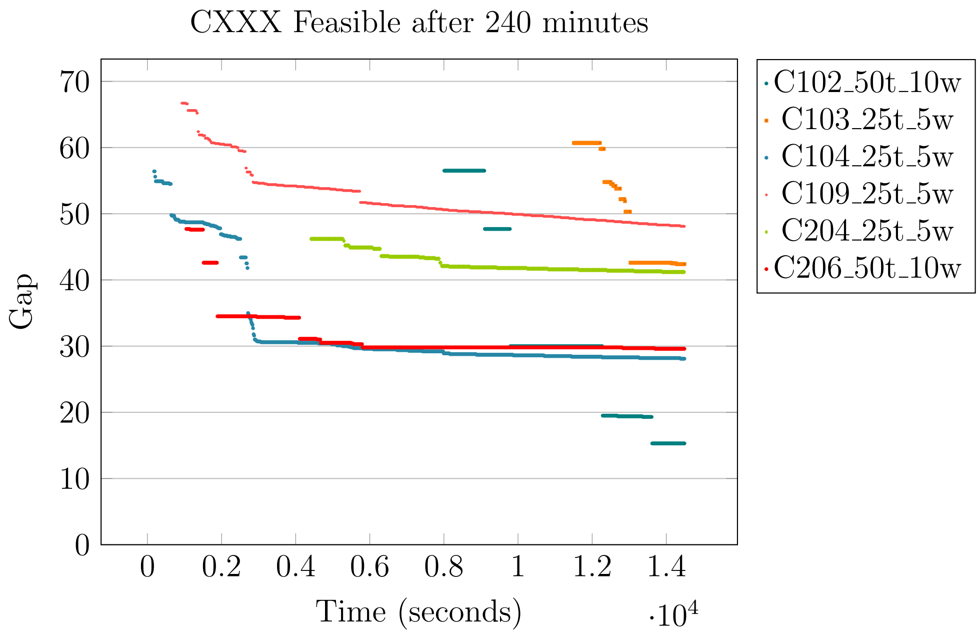

Figure B.12: Solver achieving feasible solutions for instances after increasing to a time limit of 240 minutes

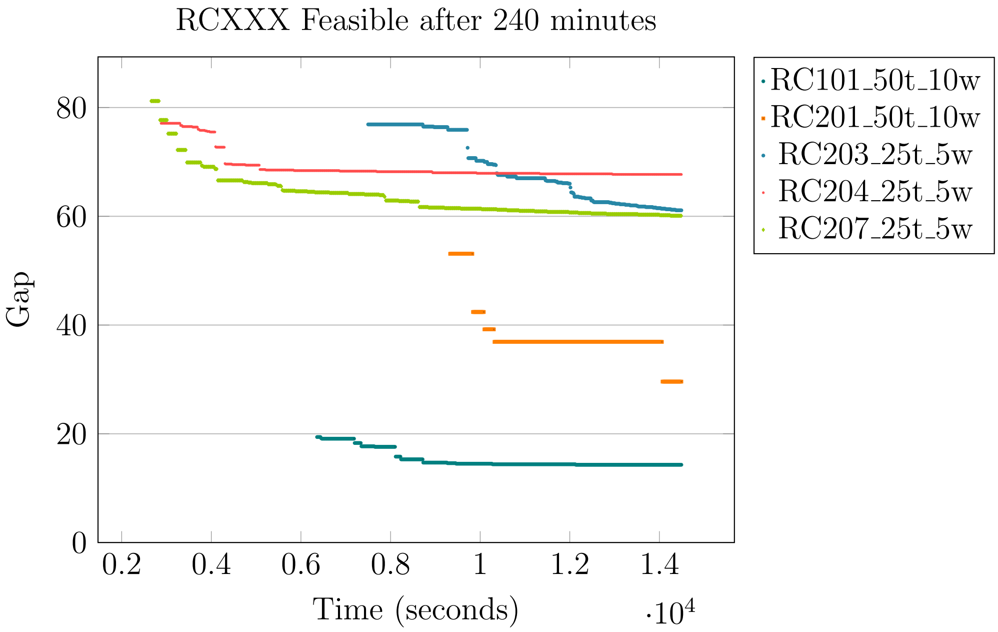

Figure B.13: Solver achieving feasible solutions for instances after increasing to a time limit of 240 minutes

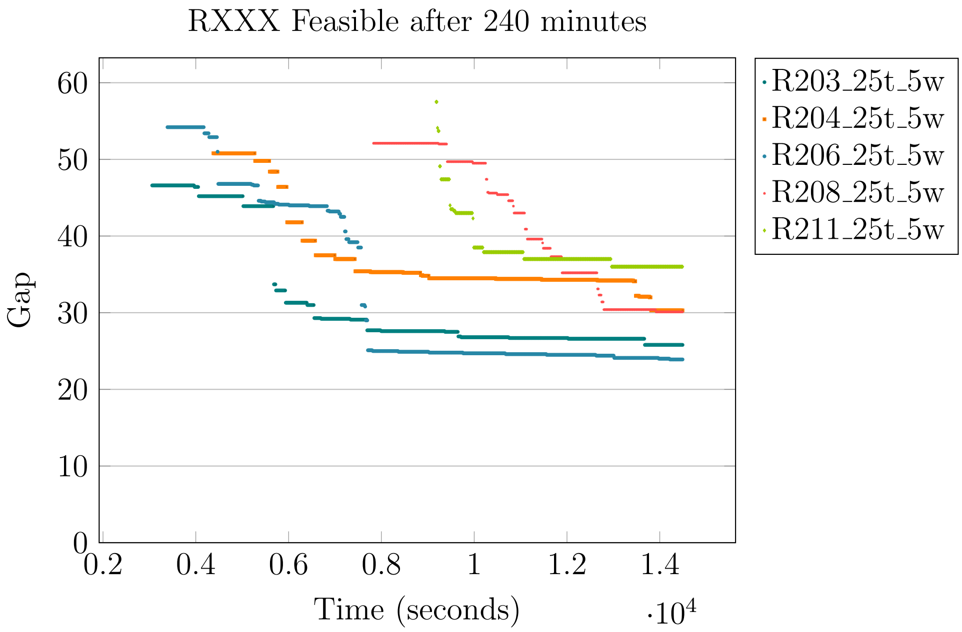

Figure B.14: Solver achieving feasible solutions for instances after increasing to a time limit of 240 minutes

Appendix C

Detailed experiments results with MILP model

Instance Name	Time	Best Obj	Bound	Gap	Category
hh_00	4356.25	6.96E+12	-6803538294	1.000977478	Non-Optimal
ll1_00	7200.11	16667174574	217961291.4	0.986922721	Non-Optimal
ll1_01	7200.06	13476948622	217961977.6	0.983827053	Non-Optimal
ll1_02	7200.07	36571184133	217960117.8	0.994040113	Non-Optimal
ll1_03	7200.2	8574920712	217962159.6	0.974581437	Non-Optimal
ll1_04	7200.03	3626168528	217962623.5	0.939891756	Non-Optimal
ll1_05	7200.06	33256410348	217957850.3	0.99344614	Non-Optimal
ll1_06	7200.06	3.04E+11	217962111	0.999282373	Non-Optimal
ll1_07	7200.07	26642544571	217957422.9	0.991819196	Non-Optimal
ll2_00	882.69	5482534825	5482456269	1.43E-05	Optimal
ll3_00	7200.07	-198615845.6	-198671925.7	0.000282354	Non-Optimal
test150-0-0-0-0_d0_tw0	204.82	-	-	0.00E+00	OutOfMemory
test150-0-0-0-0_d0_tw1	227.64	-	-	0	OutOfMemory
test150-0-0-0-0_d0_tw2	219.92	-	-	0	OutOfMemory
test150-0-0-0-0_d0_tw3	233.87	-	-	0.00E+00	OutOfMemory
test150-0-0-0-0_d0_tw4	297.67	-	-	0	OutOfMemory
test50-0-0-0-0_d0_tw0	123.61	-842599324.2	-842601771.9	2.90E-06	Optimal
test50-0-0-0-0_d0_tw1	105.81	-842599506.6	-842601114.2	1.91E-06	Optimal
test50-0-0-0-0_d0_tw2	71.51	-842599560.3	-842601056.2	1.78E-06	Optimal
test50-0-0-0-0_d0_tw3	53.73	-842598590	-842598904.6	3.73E-07	Optimal
test50-0-0-0-0_d0_tw4	257.08	-842599753.8	-842601052.4	1.54E-06	Optimal
test250-0-0-0-0_d0_tw0		-	-	-	Unloadable
test250-0-0-0-0_d0_tw1		-	-	-	Unloadable
test250-0-0-0-0_d0_tw2		-	-	-	Unloadable
test250-0-0-0-0_d0_tw3		-	-	-	Unloadable
test250-0-0-0-0_d0_tw4		-	-	-	Unloadable
BTEngineers		-	-	-	Unloadable
1_District0	7200.09	5.15E+11	-20168585306	1.039192633	Non-Optimal
1_District1	7294.02	8.15E+13	5.76E+12	9.29E-01	Non-Optimal
1_District2	7200.07	9.80E+11	46047416369	0.953023385	Non-Optimal
1_District3	7200.24	2.56E+12	-27961999838	1.01094368	Non-Optimal
1_District4	756.21	-	-	0	OutOfMemory
1_District5	415.59	-	-	0	OutOfMemory
10_District0	7200.02	71563566909	-6815574703	1.095238052	Non-Optimal
10_District1	7200.12	1.53E+13	9.39E+11	0.938766147	Non-Optimal
10_District2	7200.02	2.78E+10	-1214374567	1.043728136	Non-Optimal
10_District3	7200.05	1.75E+11	13062383793	0.925292423	Non-Optimal
10_District4	7219.8	5.81E+13	2.23E+12	0.961684005	Non-Optimal
10_District5	7203.97	7.31E+13	9.46E+12	0.870548433	Non-Optimal

Table C.1: Solver MILP results with time limit of 2 hours. Part 1

Instance Name	Time	Best Obj	Bound	Gap	Category	Instance Name	Time	Best Obj	Bound	Gap	Category
11_District0	7200.3	3046672128	-343582739.3	1.11E+00	Non-Optimal	17_District0	7200.48	60633779564	10566769884	8.26E-01	Non-Optimal
11_District1	7200.46	1.22E+13	1.23E+12	0.899394132	Non-Optimal	17_District1	7200.66	2.23E+13	2.72E+12	8.78E-01	Non-Optimal
11_District2	7200.16	9214556192	-683028187.4	1.074124914	Non-Optimal	17_District2	7200.04	1.12E+11	32674378501	7.08E-01	Non-Optimal
11_District3	7203.74	61560228131	-6724661773	1.109237116	Non-Optimal	17_District3	7200.07	2.66E+11	14798591845	9.44E-01	Non-Optimal
11_District4	7205.53	5.09E+13	2.06E+12	0.959516995	Non-Optimal	17_District4	7206.14	5.19E+13	3.08E+12	9.41E-01	Non-Optimal
11_District5	7200.27	3.10E+13	3.72E+12	0.87999207	Non-Optimal	17_District5	7203.41	3.49E+13	4.95E+12	8.58E-01	Non-Optimal
12_District0	7200.02	1.15E+11	4709971071	9.59E-01	Non-Optimal	18_District0	7200.04	2341527489	983897485.8	5.80E-01	Non-Optimal
12_District1	7232.86	-	2.87E+12	0.00E+00	No solution	18_District1	7200.82	1.84E+13	1.54E+12	9.16E-01	Non-Optimal
12_District2	7200.05	1.65E+11	15559460113	0.905567519	Non-Optimal	18_District2	7200.01	4417358464	1781087625	5.97E-01	Non-Optimal
12_District3	7200.04	4.31E+11	2461737497	0.994287514	Non-Optimal	18_District3	7200.04	73407720408	553699748.1	9.92E-01	Non-Optimal
12_District4	484.51	-	-	0.00E+00	OutOfMemory	18_District4	7200.32	-	1.45E+12	0.00E+00	No solution
12_District5	274.91	-	-	0	OutOfMemory	18_District5	7200.77	2.51E+13	2.82E+12	8.87E-01	Non-Optimal
13_District0	7200.03	1.54E+11	686439502.4	0.995551703	Non-Optimal	19_District0	7200.03	70282556413	-4120933349	1.06E+00	Non-Optimal
13_District1	7213.28	3.30E+13	2.92E+12	0.911590702	Non-Optimal	19_District1	7208.15	4.33E+13	2.41E+12	9.44E-01	Non-Optimal
13_District2	7200.04	1.49E+11	14472482331	9.03E-01	Non-Optimal	19_District2	7200.02	1.53E+11	50109469879	6.73E-01	Non-Optimal
13_District3	7208.93	8.03E+11	33722792247	9.58E-01	Non-Optimal	19_District3	7200.27	4.27E+11	843264452.5	9.98E-01	Non-Optimal
13_District4	291.53	-	-	0.00E+00	OutOfMemory	19_District4	1165.56	-	-	0.00E+00	OutOfMemory
13_District5	7230.61	9.28E+13	6.10E+12	0.934228553	Non-Optimal	19_District5	7240.76	1.63E+14	1.42E+13	9.13E-01	Non-Optimal
14_District0	7200.02	34977146773	-1558573890	1.044559778	Non-Optimal	2_District0	7200.35	2.41E+11	-9756014537	1.04E+00	Non-Optimal
14_District1	7201.05	2.66E+13	1.83E+12	9.31E-01	Non-Optimal	2_District1	7231.6	7.63E+13	5.47E+12	9.28E-01	Non-Optimal
14_District2	7200.43	9.02E+10	12629794912	0.859926711	Non-Optimal	2_District2	7200.04	3.96E+11	20197401474	9.49E-01	Non-Optimal
14_District3	7200.05	3.76E+11	23808393808	0.936608097	Non-Optimal	2_District3	7200.1	2.26E+12	8455432517	9.96E-01	Non-Optimal
14_District4	7221.33	7.85E+13	2.92E+12	9.63E-01	Non-Optimal	2_District4	7266.11	1.41E+14	6.86E+12	9.51E-01	Non-Optimal
14_District5	7222.27	-	8.27E+12	0	No solution	2_District5	7335.64	2.07E+14	1.27E+13	9.39E-01	Non-Optimal
15_District0	7200.05	42188641727	-3067386517	1.072706453	Non-Optimal	16_District0	7200.02	1.80E+11	-1556159339	1.01E+00	Non-Optimal
15_District1	7200.79	2.37E+13	1.31E+12	0.944575937	Non-Optimal	16_District1	7200.8	2.50E+13	2.40E+12	9.04E-01	Non-Optimal
15_District2	7200.05	67421894827	2957696106	9.56E-01	Non-Optimal	16_District2	7200.02	94504646914	23905291424	7.47E-01	Non-Optimal
15_District3	7200.09	9.20E+11	-16856176375	1.02E+00	Non-Optimal	16_District3	7200.04	4.46E+11	31132673293	9.30E-01	Non-Optimal
15_District4	7228.67	-	4.44E+11	0.00E+00	No solution	16_District4	7235.16	7.19E+13	1.78E+12	9.75E-01	Non-Optimal
15_District5	7215.64	6.13E+13	3.98E+12	9.35E-01	Non-Optimal	16_District5	7257.82	6.71E+13	7.10E+12	8.94E-01	Non-Optimal
20_District0	7200.02	98421186256	-1921847118	1.02E+00	Non-Optimal	26_District0	7202.11	3.63E+11	-13031447627	1.03586897	Non-Optimal
20_District1	7200.34	2.47E+13	3.09E+12	8.75E-01	Non-Optimal	26_District1	7243.59	8.22E+13	5.12E+12	0.937700298	Non-Optimal
20_District2	7200.02	27378465012	3786982803	8.62E-01	Non-Optimal	26_District2	7200.11	2.83E+11	1.27E+11	0.551892157	Non-Optimal
20_District3	7200.58	4.99E+11	1454888121	9.97E-01	Non-Optimal	26_District3	7200.02	-	1.43E+11	0	No solution
20_District4	7235.05	6.78E+13	1.32E+12	9.81E-01	Non-Optimal	26_District4	798.6	-	-	0	OutOfMemory
20_District5	7260.08	7.18E+13	7.35E+12	8.98E-01	Non-Optimal	26_District5	7320.3	-	9.88E+12	0	No solution
21_District0	7200.35	1.45E+11	-8423755388	1.06E+00	Non-Optimal	27_District0	7200.07	1.62E+11	-10048422754	1.061844177	Non-Optimal
21_District1	7201.09	3.32E+13	1.95E+12	9.41E-01	Non-Optimal	27_District1	7201.58	-	1.84E+12	0	No solution
21_District2	7200.04	62666013838	1.69E+10	7.30E-01	Non-Optimal	27_District2	7200.02	1.08E+11	-3313818927	1.030687852	Non-Optimal
21_District3	7200.1	1.45E+12	2.67E+10	0.981532757	Non-Optimal	27_District3	7200.07	1.47E+11	-4819496612	1.032762908	Non-Optimal
21_District4	7235.09	-	2.48E+12	0.00E+00	No solution	27_District4	7268.69	9.00E+13	3.15E+12	0.965032809	Non-Optimal
21_District5	7247.94	9.87E+13	5.85E+12	9.41E-01	Non-Optimal	27_District5	7226.08	5.54E+13	6.10E+12	0.89007042	Non-Optimal
22_District0	7200.05	1.39E+11	-6949679942	1.050031334	Non-Optimal	28_District0	7200.03	1.02E+11	3062493248	0.970007416	Non-Optimal
22_District1	7201.01	2.92E+13	2.46E+12	0.915863177	Non-Optimal	28_District1	7200.29	3.02E+13	2.47E+12	0.91798533	Non-Optimal
22_District2	7200.55	3.58E+11	6.81E+10	8.10E-01	Non-Optimal	28_District2	7200.05	64692968024	7259267855	0.887788919	Non-Optimal
22_District3	7200.09	4.91E+11	-1.09E+10	1.022136123	Non-Optimal	28_District3	7200.1	1.55E+12	-49204104303	1.031669863	Non-Optimal
22_District4	7235.84	-	1.70E+12	0.00E+00	No solution	28_District4	7219.21	6.27E+13	1.59E+12	0.974648174	Non-Optimal
22_District5	7221.8	8.43E+13	8.26E+12	9.02E-01	Non-Optimal	28_District5	7221.16	7.01E+13	6.19E+12	0.911706081	Non-Optimal
23_District0	7200.05	32906581132	14956836393	0.545475833	Non-Optimal	29_District0	7200.03	29353316624	2250742781	0.923322369	Non-Optimal
23_District1	7211.85	3.51E+13	2.57E+12	0.926818972	Non-Optimal	29_District1	7200.51	-	7.43E+11	0	No solution
23_District2	7200.03	1.84E+11	61957661187	0.663037911	Non-Optimal	29_District2	7200.04	1.10E+11	11858105776	0.892521525	Non-Optimal
23_District3	7200.07	3.30E+11	33876351191	0.897297985	Non-Optimal	29_District3	7200.06	6.95E+11	-9177556504	1.013207977	Non-Optimal
23_District4	7239.96	-	3.68E+12	0	No solution	29_District4	7200.52	2.14E+13	1.01E+12	0.952919132	Non-Optimal
23_District5	7380.67	-	1.47E+13	0	No solution	29_District5	217.31	-	-	0	OutOfMemory
24_District0	7200.04	1.06E+11	3273835928	0.969065212	Non-Optimal	3_District0	7200.04	9.07E+10	-4962067494	1.054707558	Non-Optimal
24_District1	7200.52	-	2.78E+12	0	No solution	3_District1	7200.98	3.24E+13	2.68E+12	0.917238226	Non-Optimal
24_District2	7200.32	26055867879	8881288495	0.659144399	Non-Optimal	3_District2	7200.07	1.89E+11	31594209651	0.833238322	Non-Optimal
24_District3	7200.69	3.07E+11	-11758446498	1.03836129	Non-Optimal	3_District3	7200.08	1.56E+12	1719905334	0.998896959	Non-Optimal
24_District4	7236.28	6.94E+13	1.71E+12	0.975304561	Non-Optimal	3_District4	7294.08	1.85E+14	6.61E+12	0.964358333	Non-Optimal

Table C.2 – continued from previous page

Instance Name	Time	Best Obj	Bound	Gap	Category	Instance Name	Time	Best Obj	Bound	Gap	Category
24_District5	7209.34	-	-3.28E+11	0	No solution	3_District5	7208.4	1.27E+14	8.56E+12	0.932814908	Non-Optimal
25_District0	7201.72	1255504790	-232811062.1	1.185432237	Non-Optimal	30_District0	7200.07	1.49E+10	4368418960	0.705960652	Non-Optimal
25_District1	7200.09	1.21E+13	2.39E+12	0.803648051	Non-Optimal	30_District1	7200.09	6.03E+12	1.72E+12	0.714441049	Non-Optimal
25_District2	7200.02	2430507031	1590155386	0.345751579	Non-Optimal	30_District2	7200.07	2.90E+10	20012134951	0.30956102	Non-Optimal
25_District3	7200.02	54825213021	7726221833	0.859075389	Non-Optimal	30_District3	7200.06	1.56E+11	7922051937	0.949105043	Non-Optimal
25_District4	7215.86	5.13E+13	2.71E+12	0.947152415	Non-Optimal	30_District4	7200.66	1.48E+13	4.97E+11	0.966441272	Non-Optimal
25_District5	7200.66	2.00E+13	1.67E+12	0.916628109	Non-Optimal	30_District5	7200.38	-	6.16E+11	0	No solution
4_District0	7200	2.04E+10	2189513570	0.892561051	Non-Optimal	C101_100t_20w	7202.05	-1.23E+10	-12548081860	0.019762868	Non-Optimal
4_District1	7200.65	2.13E+13	4.02E+12	0.811730311	Non-Optimal	C101_25t_5w	7.89	9.51E+06	9512566.64	0	Optimal
4_District2	7200.06	1.98E+10	5829765174	0.705150297	Non-Optimal	C101_50t_10w	30.83	-1.08E+09	-1.08E+09	1.76E-07	Optimal
4_District3	7200.03	3.12E+10	-1881241019	1.06021312	Non-Optimal	C102_100t_20w	7201.29	7.30E+11	-1.26E+10	1.017266662	Non-Optimal
4_District4	7239.01	-	5.61E+12	0	No solution	C102_25t_5w	7200.03	-37546115.76	-48413308.13	0.289435862	Non-Optimal
4_District5	7201.05	4.25E+13	7.35E+12	0.827063059	Non-Optimal	C102_50t_10w	7200.01	-1.10E+09	-1.12E+09	0.014134537	Non-Optimal
5_District0	7200.29	1.05E+11	7090179786	0.932670091	Non-Optimal	C103_100t_20w	7202.91	4.92E+11	-12596721561	1.025599204	Non-Optimal
5_District1	7212.83	4.41E+13	2.79E+12	0.93671232	Non-Optimal	C103_25t_5w	7200.01	-43052838.46	-54066625	0.255820219	Non-Optimal
5_District2	7200.03	4.97E+10	9297427430	0.813104381	Non-Optimal	C103_50t_10w	7200.09	-1110321792	-1.13E+09	0.021053495	Non-Optimal
5_District3	7200.13	8.81E+11	-12353228074	1.014024341	Non-Optimal	C104_100t_20w	7394.13	4.03E+11	-1.26E+10	1.031251881	Non-Optimal
5_District4	7261.58	1.51E+14	4.11E+12	0.972712102	Non-Optimal	C104_25t_5w	7200.01	-3002994.24	-54066760.99	17.00428395	Non-Optimal
5_District5	7396.5	-	2.07E+13	0	No solution	C104_50t_10w	7200.44	-1.09E+09	-1.13E+09	0.043011963	Non-Optimal
6_District0	7200.06	1.74E+11	-10474008449	1.060186964	Non-Optimal	C105_100t_20w	7200.21	-1.22E+10	-12596719216	0.036000131	Non-Optimal
6_District1	7214.24	3.70E+13	5.04E+12	0.863877685	Non-Optimal	C105_25t_5w	110.42	1.00E+06	1002123.758	6.68E-05	Optimal
6_District2	7200.07	2.64E+11	45146641624	0.828938296	Non-Optimal	C105_50t_10w	7200.07	-1102529615	-1117047107	0.01316744	Non-Optimal
6_District3	7200.06	4.61E+11	-7179533078	1.015568495	Non-Optimal	C106_100t_20w	7200.12	-4.38E+09	-1.26E+10	1.877780943	Non-Optimal
6_District4	7257.84	8.16E+13	5.02E+11	0.993855659	Non-Optimal	C106_25t_5w	9.13	9.51E+06	9512566.64	0	Optimal
6_District5	7222.02	1.11E+14	1.04E+13	0.907059836	Non-Optimal	C106_50t_10w	7200.1	-1.09E+09	-1.09E+09	0.007168325	Non-Optimal
7_District0	7200.02	7.22E+10	-4118489077	1.057053584	Non-Optimal	C107_100t_20w	7200.07	2.24E+09	-12596719712	6.630420484	Non-Optimal
7_District1	7201.96	-	5.27E+12	0	No solution	C107_25t_5w	3361.23	-2.00E+06	-2001618.6	9.91E-05	Optimal
7_District2	7200.07	1.00E+11	21483381266	0.785271898	Non-Optimal	C107_50t_10w	147.56	-1.13E+09	-1.13E+09	3.02E-07	Optimal
7_District3	7201.63	1.07E+12	-23116306380	1.021688295	Non-Optimal	C108_100t_20w	7200.17	2.03E+11	-1.26E+10	1.062132643	Non-Optimal
7_District4	7228.95	7.32E+13	2.49E+12	0.96590525	Non-Optimal	C108_25t_5w	7200.01	-8009072.98	-45555855.04	4.688030956	Non-Optimal
7_District5	7263.11	1.14E+14	1.10E+13	0.903130012	Non-Optimal	C108_50t_10w	5813.83	-1.13E+09	-1.13E+09	6.85E-07	Optimal
8_District0	7200.01	4.78E+10	-1875540161	1.039224143	Non-Optimal	C109_100t_20w	7202.6	4.18E+11	-12596721151	1.030139052	Non-Optimal
8_District1	7200.85	2.55E+13	2.80E+12	0.890335306	Non-Optimal	C109_25t_5w	7200.01	-8.01E+06	-52564787.38	5.562998143	Non-Optimal
8_District2	7203.85	7.32E+10	2.60E+10	0.64414183	Non-Optimal	C109_50t_10w	7200.07	-1.13E+09	-1.13E+09	0.006920901	Non-Optimal
8_District3	7201.51	6.83E+11	-13573330027	1.019881431	Non-Optimal	C201_100t_20w	227.85	-1.26E+10	-1.26E+10	1.62E-08	Optimal
8_District4	7241.77	6.92E+13	4.67E+12	0.932537378	Non-Optimal	C201_25t_5w	0.28	-4.56E+07	-45555930.86	3.06E-07	Optimal
8_District5	7216.55	7.93E+13	5.95E+12	0.924922522	Non-Optimal	C201_50t_10w	5.1	-1.13E+09	-1.13E+09	2.31E-08	Optimal
9_District0	7200.03	1.00E+11	1367203580	0.986340695	Non-Optimal	C202_100t_20w	7201.73	3.29E+11	-12596721048	1.038322104	Non-Optimal
9_District1	7200.7	2.20E+13	1.87E+12	0.914849983	Non-Optimal	C202_25t_5w	7200.01	-4.56E+07	-51563389.65	0.131869463	Non-Optimal
9_District2	7200.02	8.93E+10	5.59E+10	0.374159728	Non-Optimal	C202_50t_10w	5658.45	-1125905362	-1.13E+09	2.31E-07	Optimal
9_District3	7200.07	2.67E+12	3.24E+10	0.987838234	Non-Optimal	C203_100t_20w	7201.27	6.55E+11	-1.26E+10	1.019224343	Non-Optimal
9_District4	7234.97	1.01E+14	4.62E+12	0.954445987	Non-Optimal	C203_25t_5w	7200.03	-4.56E+07	-54066539.88	0.186816046	Non-Optimal
9_District5	7204.58	5.32E+13	1.23E+13	0.769261309	Non-Optimal	C203_50t_10w	4524.63	-1.13E+09	-1.13E+09	4.90E-07	Optimal
C204_100t_20w	7209.12	7.30E+11	-12596722357	1.017266664	Non-Optimal	R108_100t_20w	7213.13	54674971463	-1399634151	1.03E+00	Non-Optimal
C204_25t_5w	7200.02	-4.56E+07	-54066607.11	0.186816981	Non-Optimal	R108_25t_5w	7200.01	2.60E+07	-5727851.626	1.22001929	Non-Optimal
C204_50t_10w	7200.07	-1125905585	-1.13E+09	6.92E-03	Non-Optimal	R108_50t_10w	7200.02	293491506	-125964681.9	1.42919362	Non-Optimal
C205_100t_20w	3876.8	-1.26E+10	-1.26E+10	1.72E-07	Optimal	R109_100t_20w	7200.11	37368662167	-1399632913	1.04E+00	Non-Optimal
C205_25t_5w	50.62	-45555926.8	-45556820.17	1.96E-05	Optimal	R109_25t_5w	7200.04	2.63E+07	-4585291.753	1.174638169	Non-Optimal
C205_50t_10w	638.97	-1125905404	-1125906223	7.27E-07	Optimal	R109_50t_10w	7200.49	631133985.4	-124132411.3	1.196681551	Non-Optimal
C206_100t_20w	7202.72	2.62E+11	-12596721299	1.05E+00	Non-Optimal	R110_100t_20w	8013.98	81060000000	-1399633800	1.01726664	Non-Optimal
C206_25t_5w	600.48	-4.56E+07	-45557397.29	3.23E-05	Optimal	R110_25t_5w	7200.04	34488424.66	-5783439.701	1.17E+00	Non-Optimal
C206_50t_10w	3387.36	-1125905541	-1125917733	1.08E-05	Optimal	R110_50t_10w	7200.02	636761297.6	-125964262.1	1.20E+00	Non-Optimal
C207_100t_20w	7200.56	2.17E+11	-12596721620	1.057941823	Non-Optimal	R111_100t_20w	7204.13	46452786127	-1399633568	1.03E+00	Non-Optimal
C207_25t_5w	7200.03	-4.56E+07	-50561958.58	0.109887446	Optimal	R111_25t_5w	7200.02	2.61E+07	-5552034.885	1.21235848	Non-Optimal
C207_50t_10w	2737.48	-1125905456	-1125906658	1.07E-06	Non-Optimal	R111_50t_10w	7200.09	496942686.2	-125964289.2	1.25E+00	Non-Optimal
C208_100t_20w	7200.74	5.44E+11	-12596721654	1.023149799	Non-Optimal	R112_100t_20w	2961.6	62929580975	-1399634330	1.022241278	Non-Optimal
C208_25t_5w	5896.54	-4.56E+07	-45559966.48	8.85E-05	Optimal	R112_25t_5w	7200	2.63E+07	-6006187.135	1.228272598	Non-Optimal
C208_50t_10w	4139.53	-1125905466	-1125906151	6.09E-07	Optimal	R112_50t_10w	7200.09	4.27E+08	-1.26E+08	1.294827074	Non-Optimal
R101_100t_20w	7200.09	6319981503	4071154664	0.355828073	Non-Optimal	R201_100t_20w	7200.14	-1383419123	-1.40E+09	0.011719699	Non-Optimal

Table C.2 – continued from previous page

Instance Name	Time	Best Obj	Bound	Gap	Category	Instance Name	Time	Best Obj	Bound	Gap	Category
R101_25t_5w	0.25	47893886.58	47893886.58	0	Optimal	R201_25t_5w	31.85	-5171519.46	-5171984.127	8.99E-05	Optimal
R101_50t_10w	331.6	709051142.4	709051142.4	0.00E+00	Optimal	R201_50t_10w	26.32	-1.25E+08	-1.25E+08	4.39E-06	Optimal
R102_100t_20w	7200.18	42318726218	-575522285.5	1.013599707	Non-Optimal	R202_100t_20w	7200.95	56342105263	-1399633169	1.024841691	Non-Optimal
R102_25t_5w	7200.03	34655177	360093.6341	0.98960924	Non-Optimal	R202_25t_5w	7200.03	-5171550.18	-5894425.765	0.139779285	Non-Optimal
R102_50t_10w	7200.06	641955625.9	-55290835.21	1.086128749	Non-Optimal	R202_50t_10w	7200.07	-125097689.9	-1.26E+08	0.006924478	Non-Optimal
R103_100t_20w	7204.22	49716801838	-1399633258	1.028152118	Non-Optimal	R203_100t_20w	8949.96	8.11E+10	-1.40E+09	1.01726664	Non-Optimal
R103_25t_5w	7200.01	30260803.34	-840866.9964	1.027787332	Non-Optimal	R203_25t_5w	7200.02	-5.17E+06	-6006035.549	0.161346331	Non-Optimal
R103_50t_10w	7200.09	570531480.8	-58868675.49	1.10E+00	Non-Optimal	R203_50t_10w	7200.09	-1.25E+08	-1.26E+08	0.006927004	Non-Optimal
R104_100t_20w	7204.43	61254341207	-1399633958	1.02E+00	Non-Optimal	R204_100t_20w	7210.32	81060000000	-1399634294	1.017266646	Non-Optimal
R104_25t_5w	7200.01	3.00E+07	-5561027.288	1.18547476	Non-Optimal	R204_25t_5w	7200.03	-5.06E+06	-6006120.92	0.186874843	Non-Optimal
R104_50t_10w	7200.26	293491458.9	-58869027.45	1.20058174	Non-Optimal	R204_50t_10w	7200.09	-1.25E+08	-1.26E+08	0.006928452	Non-Optimal
R105_100t_20w	7200.81	14550273831	-1386269979	1.10E+00	Non-Optimal	R205_100t_20w	7201.56	52216151204	-1.40E+09	1.026804603	Non-Optimal
R105_25t_5w	263.85	3.52E+07	35155829.64	0	Optimal	R205_25t_5w	7200.03	-5.17E+06	-5616582.647	0.086010085	Non-Optimal
R105_50t_10w	7200.14	388290958.1	-96222194.67	1.247809517	Non-Optimal	R205_50t_10w	7200.07	-1.25E+08	-1.26E+08	0.006924347	Non-Optimal
R106_100t_20w	7207.65	27457727146	-1399632846	1.050974097	Non-Optimal	R206_100t_20w	7202.54	4.40E+10	-1399633589	1.0318279	Non-Optimal
R106_25t_5w	7200.01	30483330.46	-4912682.401	1.16E+00	Non-Optimal	R206_25t_5w	7200.04	-5.17E+06	-6005971.036	0.161289086	Non-Optimal
R106_50t_10w	7200.02	303880567.4	-125097944.4	1.41E+00	Non-Optimal	R206_50t_10w	7200.07	-1.25E+08	-1.26E+08	0.006926714	Non-Optimal
R107_100t_20w	7206.44	68698350493	-1399633479	1.02E+00	Non-Optimal	R207_100t_20w	9009.21	6.21E+10	-1.40E+09	1.022541307	Non-Optimal
R107_25t_5w	7200.01	2.20E+07	-5673932.363	1.257573715	Non-Optimal	R207_25t_5w	7200.03	-5171794.96	-6006098.673	0.161318018	Non-Optimal
R107_50t_10w	7200.02	372274723.5	-125964228.6	1.338363635	Non-Optimal	R207_50t_10w	7200.07	-125097967.8	-1.26E+08	0.006926867	Non-Optimal
R208_100t_20w	8505.04	8.11E+10	-1399634321	1.017266646	Non-Optimal	RC201_100t_20w	7210.79	-1378013036	-1399631667	0.015688263	Non-Optimal
R208_25t_5w	7200.01	-5.06E+06	-6006121.433	0.186849033	Non-Optimal	RC201_25t_5w	76.95	-5171005.4	-5171512.201	9.80E-05	Optimal
R208_50t_10w	7200.06	-1.25E+08	-1.26E+08	0.006927553	Non-Optimal	RC201_50t_10w	457.22	-125096456	-125098176.9	1.38E-05	Optimal
R209_100t_20w	7201.95	6.05E+10	-1.40E+09	1.023148731	Non-Optimal	RC202_100t_20w	7201.06	51392041422	-1399632695	1.027234425	Non-Optimal
R209_25t_5w	7200.01	-5060483.06	-5505464.432	0.087932588	Non-Optimal	RC202_25t_5w	7200.02	-5171480.42	-5727510.307	0.107518513	Non-Optimal
R209_50t_10w	7200.07	-1.25E+08	-1.26E+08	0.006925763	Non-Optimal	RC202_50t_10w	7200.07	-125096955.1	-125963429.4	0.006926422	Non-Optimal
R210_100t_20w	7343.26	29100542983	-1399633636	1.048096478	Non-Optimal	RC203_100t_20w	7317.13	81060000000	-1399633566	1.017266637	Non-Optimal
R210_25t_5w	7200.01	-5.17E+06	-5727866.756	0.107552766	Non-Optimal	RC203_25t_5w	7200.01	-5171505.38	-6006135.342	0.161390137	Non-Optimal
R210_50t_10w	7200.07	-1.25E+08	-1.26E+08	0.006926011	Non-Optimal	RC203_50t_10w	7200.06	-125096765.2	-125964453.5	0.006936137	Non-Optimal
R211_100t_20w	7211.23	8.11E+10	-1.40E+09	1.017266646	Non-Optimal	RC204_100t_20w	10219.84	61267851641	-1399634107	1.022844511	Non-Optimal
R211_25t_5w	7200.01	-5.06E+06	-6006171.316	0.186872433	Non-Optimal	RC204_25t_5w	7200.01	-5060413.36	-6006348.934	0.186928519	Non-Optimal
R211_50t_10w	7200.26	-1.23E+08	-1.26E+08	0.028276543	Non-Optimal	RC204_50t_10w	7200.09	-125097014.9	-125964909.1	0.006937769	Non-Optimal
RC101_100t_20w	7200.14	81060000000	-1345044764	1.0165932	Non-Optimal	RC205_100t_20w	7200.99	39016882589	-1399632531	1.035872485	Non-Optimal
RC101_25t_5w	200.88	2.70E+07	27034405.38	0	Optimal	RC205_25t_5w	7200.02	-5171462.06	-5338428.02	0.032286026	Non-Optimal
RC101_50t_10w	7200.09	1.90E+08	-9.64E+07	1.507395824	Non-Optimal	RC205_50t_10w	7200.07	-125096766.4	-125963284.3	0.006926781	Non-Optimal
RC102_100t_20w	7200.07	3.16E+10	-1.40E+09	1.044322616	Non-Optimal	RC206_100t_20w	7204.52	54726310289	-1399632655	1.025575133	Non-Optimal
RC102_25t_5w	7200.02	26645148.48	-5023761.64	1.188543203	Non-Optimal	RC206_25t_5w	7200.32	-5171397.94	-5616588.71	0.086087123	Non-Optimal
RC102_50t_10w	7200.1	578756223	-54724429.97	1.094555234	Non-Optimal	RC206_50t_10w	7200.07	-125097137.8	-125963744.2	0.006927468	Non-Optimal
RC103_100t_20w	7201.45	5.22E+10	-1399633064	1.0268046	Non-Optimal	RC207_100t_20w	7209.42	79446906218	-1399633363	1.017617217	Non-Optimal
RC103_25t_5w	7200.01	21805915.88	-5707482.779	1.261740108	Non-Optimal	RC207_25t_5w	7200.04	-5059890.2	-5780084.413	0.142333961	Non-Optimal
RC103_50t_10w	7200.09	767922882.8	-125098325.4	1.162904802	Non-Optimal	RC207_50t_10w	7200.15	-122499526.2	-125964374	0.028284581	Non-Optimal
RC104_100t_20w	8929.45	48892692141	-1399633535	1.028626641	Non-Optimal	RC208_100t_20w	7824.63	61281361512	-1399634248	1.022839477	Non-Optimal
RC104_25t_5w	7200.03	25810879.24	-5728139.101	1.221927314	Non-Optimal	RC208_25t_5w	7200.01	-5060744	-6006530.656	0.18688688	Non-Optimal
RC104_50t_10w	7200.7	498674346.2	-125099080.2	1.250863276	Non-Optimal	RC208_50t_10w	7200.14	-125096819.2	-125964960.6	0.006939757	Non-Optimal
RC105_100t_20w	7200.1	33231901421	-1399631733	1.042117113	Non-Optimal	RC105_25t_5w	7200.03	30761388.42	140264.0041	0.995440258	Non-Optimal
RC105_50t_10w	7200.96	448893536.2	-119991667.3	1.267305402	Non-Optimal	RC106_100t_20w	7200.15	57147301655	-1399632567	1.024491665	Non-Optimal
RC106_25t_5w	7200.01	22417682.52	-5116160.288	1.228219857	Non-Optimal	RC106_50t_10w	7200.03	515989390.6	-122398831.4	1.237211915	Non-Optimal
RC107_100t_20w	7201.4	55499082255	-1399633328	1.025219036	Non-Optimal	RC107_25t_5w	7200.03	30761589.88	-5916392.185	1.192330507	Non-Optimal
RC107_50t_10w	7200.09	575726019.2	-125964519.7	1.218792473	Non-Optimal	RC108_100t_20w	7207.19	62110875294	-1399634218	1.022534447	Non-Optimal
RC108_25t_5w	7200.01	30372205.56	-6006196.565	1.197753059	Non-Optimal	RC108_50t_10w	7200.04	630268451.2	-125964908.6	1.199859137	Non-Optimal

Table C.2: Solver MILP results with time limit of 2 hours. Part 2

Appendix D

Greedy Heuristic's Pseudo-code and Results

D.1 GH2's pseudo-code

Algorithm 14 Greedy Heuristic

```
procedure SOLVE
    visitList ← copy of visits(V)
    sol ← CREATESOLUTIONSTRUCTURE
    SORT(visitList, listCriterion)
    while visitList is not empty do
        SORT(sol, solCriterion)
        v ← visitList.remove(0)
        candidates ← ALLOCPOSSIBLEANY(v,sol)
        SORT(candidates)
        if candidates is not empty then
            c ← candidates.remove(0)
            INCLUDE(c,sol)
            i ← v.required
            for i > 1 do
                if candidates is not empty then
                    c ← candidates.remove(0)
                    INCLUDE(c,sol)
                else
                    UNALLOCATE(v,sol)
        else
            UNALLOCATE(v,sol)
```

D.2 Creation of Catalogue Index

Algorithm 15 Greedy Heuristic

```
function CatalogueIndex(can)
    datesC
    dcList
    for c ← 1, can do
        if ¬datesC.contains(c.st) then
            datesC.add(c.st)
            dc ← NewDC(c.st,NewCC())
            dcList.add(dc)
        sft = c.st + c.ft
        if ¬datesC.contains(sft) then
            datesC.add(sft)
            dc ← NewDC(sft,NewCC())
            dcList.add(dc)
    Sort(dcList, criterion)
    for dc ← 1, dcList do
        for c ← 1, can do
            sft = c.st + c.ft
            if c.st ≤ dc.t AND dc.t ≤ sft then
                dc.cover.add(c)
    Sort(dcList, criterion2)
    return dcList
```

D.3 Results of Experiments with GH1

Table D.1: GH1 Results

Instance	Time	Objective Value	Instance	Time	Objective Value
10_District0	9	106721973623.64	24_District4	2	46587487541375.5
10_District1		-	24_District5	1	28476844997125.1
10_District2	1	103035047948.19	25_District0	0	2777463894.29
10_District3	1	216825800383.27	25_District1	1	6857920874061.2
10_District4	5	33561788181624	25_District2	0	4534524985.78
10_District5	4	40627133504499.9	25_District3	0	80069099613.02
11_District0	0	5544047563.69	25_District4	2	22926690235187.3
11_District1	2	7468316424982.42	25_District5	1	12848404341628.7
11_District2	0	15786909708.89	26_District0	1	292395405901.65
11_District3	0	138694820557.4	26_District1	2	43275522185650.6
11_District4	2	25877813149871.6	26_District2	1	396575268652.81
11_District5	1	17263020337689.7	26_District3	1	1673764458720.69
12_District0	1	153991002951.61	26_District4	3	165750615422004
12_District1	2	47778556920073.9	26_District5	2	119939068715765
12_District2	0	236235562224.57	27_District0	0	191801069899.63
12_District3	1	416169432413.25	27_District1	2	18782623767054
12_District4	3	82057857817025.1	27_District2	1	119427985194.55
12_District5	2	72805935848959.7	27_District3	1	188834841381.43
13_District0	0	200168900258.5	27_District4	3	45350677434646.8
13_District1	1	20270743939483.3	27_District5	3	46639259653995.7
13_District2	1	199199456003.19	28_District0	0	140256552967.31
13_District3	0	684785856644.61	28_District1	1	15128335139531.3
13_District4	1	48134040829393.7	28_District2	0	108216971983.04
13_District5	2	51261023782563.7	28_District3	1	1122934685459.85
14_District0	0	44356736598.4	28_District4	2	25391011993117.2
14_District1	2	14279066295349	28_District5	2	34767981234314.1
14_District2	0	146362281680.48	29_District0	0	39806924540.87
14_District3	0	403328233660.47	29_District1	1	10110720294717
14_District4		-	29_District2	0	147173272757.18
14_District5	2	56647322729362.1	29_District3	1	370198923550.63

Table D.1 – continued from previous page

Instance	Time	Objective Value	Instance	Time	Objective Value
15_District0	0	68541235610.88	29_District4	1	9513034912449
15_District1	2	16766645678524.1	29_District5	2	53512720963734.6
15_District2		-	2_District0		-
15_District3	0	613230685180.04	2_District1	2	50025849298076.5
15_District4	2	44052689348565.8	2_District2	0	309401067266.24
15_District5	2	34235585575181.9	2_District3	0	1311902709882.22
16_District0	1	138582500431.33	2_District4	3	88397409789402.3
16_District1	1	16335857606068.6	2_District5		-
16_District2	0	132514327184.6	30_District0	1	20052123066.46
16_District3	0	361273702860.36	30_District1	1	4312691699679.27
16_District4	2	43707840624537.3	30_District2	0	61722485384.6
16_District5	3	54000844649128.2	30_District3	1	182562266602.39
17_District0	0	78319859061.15	30_District4	1	8660584843183.65
17_District1	2	12432742277333.8	30_District5		-
17_District2	0	156241612427.41	3_District0	0	125936823583.02
17_District3	0	304821943175.79	3_District1	1	24220848064971.8
17_District4		-	3_District2	1	240616217504.89
17_District5	2	26566194850082.5	3_District3	1	1285270273951.59
18_District0	0	3521808556.5	3_District4	3	96234574882611.1
18_District1		-	3_District5	3	64265301930963.4
18_District2	1	9256632018.73	4_District0	0	25573824722.49
18_District3	1	100695778556.15	4_District1	2	17238235759616.8
18_District4		-	4_District2	0	28124793618.51
18_District5	2	13890094463962.3	4_District3	0	71427539474.97
19_District0	0	64625776234.49	4_District4	2	67460528170599.7
19_District1	1	25137028788637.1	4_District5		-
19_District2	1	205915841598.43	5_District0	1	129118698669.78
19_District3	1	338591139872.36	5_District1	2	22882658744118
19_District4	3	172228234833813	5_District2	1	95107414309.81
19_District5	3	93828514391975	5_District3	1	525169193508.74
1_District0	1	518609006951.86	5_District4	2	68243049339872.1
1_District1	3	56270206789859.1	5_District5	3	136613775471593
1_District2	1	1046747366809	6_District0	0	220433136825.25
1_District3	1	1560925188299.38	6_District1	1	29317596294235.9
1_District4	4	110465401558176	6_District2	0	356181224330.19
1_District5	3	99000141536068	6_District3	0	366172922416.33
20_District0	0	113316045933.83	6_District4	3	48108767553106.6
20_District1	2	17718690639232.5	6_District5	3	76979349076598.5
20_District2	0	54179107008.82	7_District0	0	63608972382.16
20_District3	1	470862883618.9	7_District1	1	26968214676885.6
20_District4	2	31749522139867.1	7_District2	0	112061719743.98
20_District5	3	59341256500235.7	7_District3	0	1074693735356.06
21_District0	0	142729859125.56	7_District4	2	51647639084098.9
21_District1	2	19809611515822.7	7_District5	3	62649939536594.5
21_District2	0	120541044419.74	8_District0	0	78001907952.28
21_District3	0	1270458349614.08	8_District1	1	13859124871569
21_District4	2	34448190397649.6	8_District2	0	90830953195.68
21_District5	2	72662216779674.1	8_District3	0	702389046706.46
22_District0	0	165570512287.3	8_District4	2	42415787308619.6
22_District1	2	15456817219610.8	8_District5	2	47359269549334.2
22_District2	1	323706119166.79	9_District0	0	120339564622.06
22_District3	1	4611956784641.1	9_District1	2	12395335811834.3
22_District4	2	48946654140257.1	9_District2	0	132520659416.61
22_District5	3	42772519577575.2	9_District3	0	1462182131302.57
23_District0	0	40743361287.58	9_District4		-
23_District1	1	17211576588480.4	9_District5	1	27216479042294
23_District2	0	248844083189.61	C101_100t_20w		-
23_District3	0	366148850953.41	C101_25t_5w	1	280350869.62
23_District4	2	61286017189336.3	C101_50t_10w	0	17324957420.36
23_District5	3	81252454364927.6	C102_100t_20w		-
24_District0	0	156052500888.05	C102_25t_5w	0	405506941.18
24_District1	1	13794367786043.8	C102_50t_10w	0	9248809635.78
24_District2	0	40759482611.7	C103_100t_20w	1	356355976237.04
24_District3	0	226783604681.59	C103_25t_5w	0	555694297.66
C103_50t_10w	0	14504344424.8	R206_100t_20w		-
C104_100t_20w		-	R206_25t_5w	0	32653362.6
C104_25t_5w	0	602753002.3	R206_50t_10w	1	880904290.06
C104_50t_10w	0	21439001405.22	R207_100t_20w		-
C105_100t_20w		-	R207_25t_5w	0	53512543
C105_25t_5w	0	709385932.94	R207_50t_10w		-
C105_50t_10w	1	16724992695.72	R208_100t_20w	1	52586327657.64
C106_100t_20w	0	106512846459.84	R208_25t_5w	0	66806535.9
C106_25t_5w	1	283354540.84	R208_50t_10w	0	2319778839.98
C106_50t_10w	1	16783430774.24	R209_100t_20w		-
C107_100t_20w		-	R209_25t_5w	0	43610939.08
C107_25t_5w	0	863578364.68	R209_50t_10w		-
C107_50t_10w		-	R210_100t_20w		-
C108_100t_20w		-	R210_25t_5w	0	15020561.04
C108_25t_5w	1	317897559.88	R210_50t_10w	0	1074399277.22
C108_50t_10w	0	16771743195.74	R211_100t_20w		-
C109_100t_20w		-	R211_25t_5w	1	53233953.8
C109_25t_5w	0	442553166.58	R211_50t_10w		-
C109_50t_10w		-	RC101_100t_20w		-
C201_100t_20w		-	RC101_25t_5w	0	58017439.38
C201_25t_5w	0	559198778.92	RC101_50t_10w	0	1415935917.12
C201_50t_10w		-	RC102_100t_20w		-
C202_100t_20w		-	RC102_25t_5w	0	52788685.5
C202_25t_5w	0	674342359.94	RC102_50t_10w	0	1602938020.88
C202_50t_10w		-	RC103_100t_20w	1	41932341565.44
C203_100t_20w	1	289821930083.4	RC103_25t_5w	0	69976782
C203_25t_5w	0	674342359.94	RC103_50t_10w	0	1937983134.7
C203_50t_10w	1	12731721881.48	RC104_100t_20w	1	57401291469.98
C204_100t_20w		-	RC104_25t_5w	0	79099300.64
C204_25t_5w	0	558197474.02	RC104_50t_10w	0	2521498398.18
C204_50t_10w	0	20897475267.18	RC105_100t_20w	1	39500541851.52
C205_100t_20w		-	RC105_25t_5w	0	57516880.62
C205_25t_5w	0	559198778.92	RC105_50t_10w	0	1869588776.72

Table D.1 – continued from previous page

Instance	Time	Objective Value	Instance	Time	Objective Value
C205_50t_10w		-	RC106_100t_20w		-
C206_100t_20w		-	RC106_25t_5w	1	62300598.44
C206_25t_5w	0	599248767.6	RC106_50t_10w		-
C206_50t_10w		-	RC107_100t_20w		-
C207_100t_20w		-	RC107_25t_5w	0	62467451.12
C207_25t_5w		-	RC107_50t_10w	1	1944476079.64
C207_50t_10w		-	RC108_100t_20w		-
C208_100t_20w		-	RC108_25t_5w	0	78932429.72
C208_25t_5w	0	599248767.6	RC108_50t_10w	0	2517602451.86
C208_50t_10w		-	RC201_100t_20w		-
R101_100t_20w	0	47671389090.02	RC201_25t_5w	0	15354397.24
R101_25t_5w	0	62467584.18	RC201_50t_10w	0	-56268456.68
R101_50t_10w	0	1540603888	RC202_100t_20w		-
R102_100t_20w	1	29848997660.04	RC202_25t_5w	0	26980073.22
R102_25t_5w	0	65471265.52	RC202_50t_10w	0	734593781.94
R102_50t_10w	0	1672630795.14	RC203_100t_20w		-
R103_100t_20w	1	42742941229.48	RC203_25t_5w	0	44612948.06
R103_25t_5w	0	65749425.48	RC203_50t_10w	0	1471345609.22
R103_50t_10w	0	1870454483.16	RC204_100t_20w	0	53399630500.72
R104_100t_20w	1	58174062614.84	RC204_25t_5w	0	61689634.12
R104_25t_5w	1	70088173.92	RC204_50t_10w	0	2323242384.8
R104_50t_10w	0	2448342328.32	RC205_100t_20w		-
R105_100t_20w		-	RC205_25t_5w	0	19026136.8
R105_25t_5w	0	62189418.38	RC205_50t_10w		-
R105_50t_10w	1	1544066777.86	RC206_100t_20w		-
R106_100t_20w		-	RC206_25t_5w	0	19637230.36
R106_25t_5w	0	65471265.52	RC206_50t_10w		-
R106_50t_10w	0	1661376059.4	RC207_100t_20w		-
R107_100t_20w		-	RC207_25t_5w	0	44946231.22
R107_25t_5w	0	65749425.48	RC207_50t_10w		-
R107_50t_10w	0	1872618948.36	RC208_100t_20w		-
R108_100t_20w	1	58195678734.76	RC208_25t_5w	1	57461940.38
R108_25t_5w	0	74371179.42	RC208_50t_10w	0	1344513753.98
R108_50t_10w	0	2582966523.96	hh_00_P0	1	4641366275065.9
R109_100t_20w		-	ll1_00_P0	1	212221208244.7
R109_25t_5w	1	70088098.08	ll1_01_P0	0	212221208244.7
R109_50t_10w		-	ll1_02_P0	1	210571622994.7
R110_100t_20w		-	ll1_03_P0	0	212221208244.7
R110_25t_5w	0	70087967.32	ll1_04_P0	1	212221208244.7
R110_50t_10w	1	2060486256.6	ll1_05_P0	0	230413363380.24
R111_100t_20w		-	ll1_06_P0	0	399713415096.83
R111_25t_5w	0	70588638.52	ll1_07_P0	0	203973281994.7
R111_50t_10w	0	2258743255.54	ll2_00_P0	0	13188111647.65
R112_100t_20w		-	ll3_00_P0	0	12012194578.59
R112_25t_5w	0	78487214.58	test150-0-0-0-0_d0_tw0		-
R112_50t_10w	0	2199006275.06	test150-0-0-0-0_d0_tw1	4	2414178385246.9
R201_100t_20w		-	test150-0-0-0-0_d0_tw2	4	2414178385246.9
R201_25t_5w	0	11849763.9599999	test150-0-0-0-0_d0_tw3	4	2414178385246.9
R201_50t_10w	0	-53239122.3	test150-0-0-0-0_d0_tw4		-
R202_100t_20w		-	test250-0-0-0-0_d0_tw0		-
R202_25t_5w	0	33098359.28	test250-0-0-0-0_d0_tw1	8	36940587325055.6
R202_50t_10w		-	test250-0-0-0-0_d0_tw2	11	36940587325055.6
R203_100t_20w	0	35585345050.76	test250-0-0-0-0_d0_tw3	9	36940587325055.6
R203_25t_5w	0	53345561.86	test250-0-0-0-0_d0_tw4		-
R203_50t_10w		-	test50-0-0-0-0_d0_tw0		-
R204_100t_20w	0	52532287550.44	test50-0-0-0-0_d0_tw1	0	-324551544.9
R204_25t_5w	0	66639648.32	test50-0-0-0-0_d0_tw2	0	-324551544.9
R204_50t_10w	0	2319778839.98	test50-0-0-0-0_d0_tw3	0	-324551544.9
R205_100t_20w		-	test50-0-0-0-0_d0_tw4		-
R205_25t_5w	0	70088211.18	BTEngineers	13	-621279823668.827
R205_50t_10w		-			

D.4 Results of Experiments with GH2

Table D.2: GH2 Results

Instance	Time	Objective Value	Instance	Time	Objective Value
10_District0	11	106721973623.64	24_District4		-
10_District1		-	24_District5	2	28462936892113.4
10_District2	1	84414256901.21	25_District0	0	2777463894.29
10_District3	1	215361002429.25	25_District1	1	7068465513197.56
10_District4	6	32819765440259.4	25_District2	1	4534524985.78
10_District5	5	40627133504499.9	25_District3	0	80069099613.02
11_District0	0	5544047563.69	25_District4	3	22928133663324
11_District1	3	7250792645874.36	25_District5	1	12516388257408.9
11_District2	1	15786909708.89	26_District0	0	292395405901.65
11_District3		-	26_District1	2	43275522185650.6
11_District4	3	25898123127639	26_District2	1	396575268652.81
11_District5	3	17263020337689.7	26_District3	1	1673764458720.69
12_District0	1	153991002951.61	26_District4		-
12_District1	3	42653795772775.5	26_District5		-
12_District2	1	236235562224.57	27_District0	0	191801069899.63
12_District3	1	474658355763.47	27_District1	1	18782623767054
12_District4		-	27_District2	0	97943182274.39
12_District5	3	72805935848959.7	27_District3	1	188834841381.43
13_District0	0	200168900258.5	27_District4		-
13_District1		-	27_District5	3	44153166761296.4

Table D.2 – continued from previous page

Instance	Time	Objective Value	Instance	Time	Objective Value
13_District2	1	186654387887.44	28_District0	1	140256552967.31
13_District3	0	667791171578.81	28_District1	2	15128335139531.3
13_District4		-	28_District2	0	108216971983.04
13_District5	3	50126639356006	28_District3	1	1077830919378.16
14_District0	0	44443988566.81	28_District4	3	25391011993117.2
14_District1		-	28_District5	3	34767981234314.1
14_District2	1	136473342260.76	29_District0	0	39806924540.87
14_District3	1	403328233660.47	29_District1		-
14_District4		-	29_District2	1	137762852813.06
14_District5	3	56649475473978.8	29_District3	0	370198923550.63
15_District0	1	68541235610.88	29_District4	1	9513034912449
15_District1	2	15294139899370.5	29_District5		-
15_District2		-	2_District0		-
15_District3	1	613230685180.04	2_District1	3	45291384772704.2
15_District4		-	2_District2	1	345750973464.34
15_District5	3	32287953051783.4	2_District3	1	1127228274524.8
16_District0	0	138477941427.15	2_District4		-
16_District1	2	15144931153129.5	2_District5		-
16_District2	1	132514327184.6	30_District0	0	20052123066.46
16_District3	1	361273702860.36	30_District1	1	4312691699679.27
16_District4		-	30_District2	1	57117742210.52
16_District5	3	51578169889298.7	30_District3	0	182562266602.39
17_District0	0	78319859061.15	30_District4	1	8052008671222.14
17_District1	1	12432742277333.8	30_District5		-
17_District2	0	156665277429.91	3_District0	1	125813335834.03
17_District3	1	279813602971.54	3_District1	2	22284769406048.7
17_District4		-	3_District2	0	240105020726.05
17_District5	2	26566194850082.5	3_District3	0	1283246863961.54
18_District0	0	3521808556.5	3_District4	4	78588143048537.1
18_District1		-	3_District5	3	64265301930963.4
18_District2	0	9256632018.73	4_District0	0	25573824722.49
18_District3	1	100695778556.15	4_District1	1	17238235759616.8
18_District4		-	4_District2	0	28124793618.51
18_District5	2	13672716327594.6	4_District3	1	60537109284.43
19_District0	0	64625776234.49	4_District4	3	64872717206335.6
19_District1	2	24528896687922	4_District5		-
19_District2	1	205915841598.43	5_District0	0	129118698669.78
19_District3	0	338591139872.36	5_District1	2	22578812619408.6
19_District4	4	123681507234826	5_District2	1	95107414309.81
19_District5	3	93828514391975	5_District3	1	525169193508.74
1_District0	0	518609006951.86	5_District4	3	68258766631008.6
1_District1	3	56270206789859.1	5_District5	4	136609756533188
1_District2		-	6_District0	0	220433136825.25
1_District3	1	1531887721228.56	6_District1	2	27537517766911.2
1_District4	4	84047639278528.5	6_District2	0	356181224330.19
1_District5	4	99000141536068	6_District3	0	363804468155.21
20_District0	0	113316045933.83	6_District4		-
20_District1	2	17718690639232.5	6_District5	3	69963236402468.5
20_District2	0	54179107008.82	7_District0	1	63608972382.16
20_District3	1	470862883618.9	7_District1	2	26968214676885.6
20_District4	3	31749522139867.1	7_District2	0	112061719743.98
20_District5	3	57561964160362.7	7_District3	1	1010758653768.06
21_District0	0	142729859125.56	7_District4		-
21_District1	2	18618763265051.4	7_District5	3	69720921596603.2
21_District2	1	116023835389.88	8_District0	0	78001907952.28
21_District3	0	1270458349614.08	8_District1	2	13859124871506.7
21_District4	3	29067095756504.5	8_District2	0	90830953195.68
21_District5		-	8_District3	0	673934851369.98
22_District0	0	165570512287.3	8_District4	2	36566921051403.7
22_District1	2	15470246253763.4	8_District5	2	46889079939457.2
22_District2	0	323706119166.79	9_District0	0	120339564622.06
22_District3	0	461195678464.1	9_District1	2	12395335811834.3
22_District4		-	9_District2	1	132353194387.96
22_District5	3	42772519577575.2	9_District3	1	1397126467231.04
23_District0	0	40743361287.58	9_District4		-
23_District1	2	17211576588480.4	9_District5	2	35072281647067.8
23_District2	0	248844083189.61	C101_100t_20w		-
23_District3	1	366148850953.41	C101_25t_5w	0	280350869.62
23_District4		-	C101_50t_10w		-
23_District5	3	80426668069628.3	C102_100t_20w		-
24_District0	0	156052500888.05	C102_25t_5w	0	330413205.7
24_District1	2	13601722471274.8	C102_50t_10w		-
24_District2	0	38666535939.24	C103_100t_20w		-
24_District3	0	226783604681.59	C103_25t_5w	0	327409315.8
C103_50t_10w		-	R206_100t_20w		-
C104_100t_20w		-	R206_25t_5w	0	32653362.6
C104_25t_5w	0	328410628.68	R206_50t_10w		-
C104_50t_10w		-	R207_100t_20w		-
C105_100t_20w		-	R207_25t_5w	0	53512543
C105_25t_5w	0	288360633.06	R207_50t_10w		-
C105_50t_10w		-	R208_100t_20w		-
C106_100t_20w	1	99144492565.14	R208_25t_5w	0	66806535.9
C106_25t_5w	0	283354540.84	R208_50t_10w		-
C106_50t_10w		-	R209_100t_20w		-
C107_100t_20w		-	R209_25t_5w		-
C107_25t_5w		-	R209_50t_10w		-
C107_50t_10w		-	R210_100t_20w		-
C108_100t_20w		-	R210_25t_5w	0	15020561.04
C108_25t_5w	0	317897559.88	R210_50t_10w		-
C108_50t_10w		-	R211_100t_20w		-
C109_100t_20w		-	R211_25t_5w	0	53233953.8
C109_25t_5w	0	442553166.58	R211_50t_10w		-
C109_50t_10w		-	RC101_100t_20w		-
C201_100t_20w		-	RC101_25t_5w	1	58017439.38
C201_25t_5w		-	RC101_50t_10w	1	1415935917.12
C201_50t_10w		-	RC102_100t_20w		-
C202_100t_20w		-	RC102_25t_5w	0	53790154.84

Table D.2 – continued from previous page

Instance	Time	Objective Value	Instance	Time	Objective Value
C202_25t_5w		-	RC102_50t_10w		-
C202_50t_10w		-	RC103_100t_20w		-
C203_100t_20w		-	RC103_25t_5w	0	58406956.56
C203_25t_5w		-	RC103_50t_10w		-
C203_50t_10w		-	RC104_100t_20w		-
C204_100t_20w		-	RC104_25t_5w	0	79433035.18
C204_25t_5w	0	558197474.02	RC104_50t_10w		-
C204_50t_10w		-	RC105_100t_20w		-
C205_100t_20w		-	RC105_25t_5w	0	49284339.76
C205_25t_5w		-	RC105_50t_10w		-
C205_50t_10w		-	RC106_100t_20w		-
C206_100t_20w		-	RC106_25t_5w	0	62300598.44
C206_25t_5w		-	RC106_50t_10w		-
C206_50t_10w		-	RC107_100t_20w		-
C207_100t_20w		-	RC107_25t_5w	0	62467451.12
C207_25t_5w		-	RC107_50t_10w		-
C207_50t_10w		-	RC108_100t_20w		-
C208_100t_20w		-	RC108_25t_5w	0	70088017.62
C208_25t_5w		-	RC108_50t_10w	1	1860931142.34
C208_50t_10w		-	RC201_100t_20w		-
R101_100t_20w	1	31324289857.8	RC201_25t_5w	0	2227025.82
R101_25t_5w	0	62467584.18	RC201_50t_10w	0	-56268456.68
R101_50t_10w	1	1480434141.18	RC202_100t_20w		-
R102_100t_20w		-	RC202_25t_5w	0	26980073.22
R102_25t_5w	0	57961949.66	RC202_50t_10w		-
R102_50t_10w		-	RC203_100t_20w		-
R103_100t_20w		-	RC203_25t_5w	0	44612948.06
R103_25t_5w	0	65749425.48	RC203_50t_10w		-
R103_50t_10w	0	1406412643.2	RC204_100t_20w		-
R104_100t_20w		-	RC204_25t_5w	0	61689634.12
R104_25t_5w	0	70088173.92	RC204_50t_10w		-
R104_50t_10w	0	2061785102.36	RC205_100t_20w		-
R105_100t_20w		-	RC205_25t_5w	0	19026136.8
R105_25t_5w	0	61299289.8	RC205_50t_10w		-
R105_50t_10w	0	1544066777.86	RC206_100t_20w		-
R106_100t_20w		-	RC206_25t_5w	0	19637230.36
R106_25t_5w	0	58406946.24	RC206_50t_10w		-
R106_50t_10w		-	RC207_100t_20w		-
R107_100t_20w		-	RC207_25t_5w		-
R107_25t_5w	0	58406920.82	RC207_50t_10w		-
R107_50t_10w	1	1483464446.74	RC208_100t_20w		-
R108_100t_20w		-	RC208_25t_5w	0	57461940.38
R108_25t_5w	0	57961903.58	RC208_50t_10w		-
R108_50t_10w	0	2000749736.76	hh_00_P0	2	4641366275065.9
R109_100t_20w		-	ll1_00_P0	1	111689842477.59
R109_25t_5w	0	61466184.98	ll1_01_P0	1	80487807895.6
R109_50t_10w		-	ll1_02_P0	1	111689842477.59
R110_100t_20w		-	ll1_03_P0	1	111689842477.59
R110_25t_5w	0	66027352.7	ll1_04_P0	1	111689842477.59
R110_50t_10w	0	1665271584.06	ll1_05_P0	1	115191304751.38
R111_100t_20w		-	ll1_06_P0		-
R111_25t_5w	0	53511883.66	ll1_07_P0	1	108592963751.38
R111_50t_10w	0	1756175567.56	ll2_00_P0	0	11461556706.86
R112_100t_20w		-	ll3_00_P0	1	6962257738.24
R112_25t_5w	0	78487214.58	test150-0-0-0-0_d0_tw0		-
R112_50t_10w	1	2064815105.06	test150-0-0-0-0_d0_tw1		-
R201_100t_20w		-	test150-0-0-0-0_d0_tw2		-
R201_25t_5w	0	1726182.96	test150-0-0-0-0_d0_tw3		-
R201_50t_10w	1	-53239122.3	test150-0-0-0-0_d0_tw4		-
R202_100t_20w		-	test250-0-0-0-0_d0_tw0		-
R202_25t_5w	0	33098359.28	test250-0-0-0-0_d0_tw1		-
R202_50t_10w		-	test250-0-0-0-0_d0_tw2		-
R203_100t_20w		-	test250-0-0-0-0_d0_tw3		-
R203_25t_5w	0	53345561.86	test250-0-0-0-0_d0_tw4		-
R203_50t_10w		-	test50-0-0-0-0_d0_tw0		-
R204_100t_20w		-	test50-0-0-0-0_d0_tw1	1	-324551544.9
R204_25t_5w	0	66639648.32	test50-0-0-0-0_d0_tw2	0	-324551544.9
R204_50t_10w		-	test50-0-0-0-0_d0_tw3	1	-324551544.9
R205_100t_20w		-	test50-0-0-0-0_d0_tw4		-
R205_25t_5w		-	BTEngineers	12	-661041733488.256
R205_50t_10w		-			

D.5 Results of Experiments with GH3

Table D.3: GH3 Results

Instance	Time	Objective Value	Instance	Time	Objective Value
10_District0	9	106514181491.02	24_District4	2	26737166769864.2
10_District1	16	9001482305057.62	24_District5	2	28479935686410
10_District2	1	58625462521.26	25_District0	1	2795528712.59
10_District3	1	209751898948.67	25_District1	1	6324900485756.72
10_District4	6	23282973667201.7	25_District2	0	4472855514
10_District5	4	40589090305550	25_District3	1	70786983265
11_District0	1	5544047418.77	25_District4	2	22271373878426.7
11_District1	3	7039742788663.01	25_District5	2	11858978750864.1
11_District2	0	16733934535.51	26_District0	0	291402864346.59
11_District3	1	94943748043.2	26_District1	2	43337058654900.8

Table D.3 – continued from previous page

Instance	Time	Objective Value	Instance	Time	Objective Value
11_District4	3	25566635276064.4	26_District2	1	432405593956.62
11_District5	2	17249049875800.2	26_District3	1	1711555856290.13
12_District0	1	153991002968.62	26_District4	5	116616244902672
12_District1	3	39368813493489.5	26_District5	4	88864369188666.8
12_District2	0	235123516633.36	27_District0	1	196822644722.21
12_District3	1	386856589843.07	27_District1	2	19532184104778.9
12_District4	4	59129219661289.8	27_District2	0	97204084369.6
12_District5	3	76568915156315	27_District3	0	188834841381.43
13_District0	1	200168900258.5	27_District4	3	34983506435735
13_District1	2	16442120749780.1	27_District5	3	43125961055732.5
13_District2	1	193620112936.58	28_District0	0	139949789802.05
13_District3	1	698181667823.88	28_District1	2	15350085980159.1
13_District4	3	34000815348145.4	28_District2	0	104194611485.93
13_District5	3	50699859698755.2	28_District3	1	1076992212853.09
14_District0	0	45850927115.85	28_District4	2	25831106950399.8
14_District1	1	14254826758086.1	28_District5	2	36105072571695.8
14_District2	1	131633438179.04	29_District0	0	39806924540.87
14_District3	0	403729007063.39	29_District1	1	7978378039376.83
14_District4	3	35559983065795.8	29_District2	0	131489239054.6
14_District5	3	53900420556895.7	29_District3	1	370499572552.62
15_District0	1	66146782173.42	29_District4	1	9184153773889.73
15_District1	1	12711526556580.4	29_District5	3	43202011804846
15_District2	0	119992921511.48	2_District0	0	203347903444.68
15_District3	1	612534148647.44	2_District1	3	40571278712927
15_District4	3	26167980969582	2_District2	0	281947676352.62
15_District5	2	32316119579039.1	2_District3	0	1178651432555.58
16_District0	1	134797464599.05	2_District4	4	68080523243923
16_District1	1	13544030449927.9	2_District5	4	112807731538713
16_District2	0	136369984551.11	30_District0	0	20070374480.49
16_District3	0	310893633570.54	30_District1	1	4312691699679.27
16_District4	3	32015923640900.9	30_District2	1	55209612841.85
16_District5	3	49863031314372.5	30_District3	0	181689072414.71
17_District0	0	80339019701.64	30_District4	1	6693326983564.53
17_District1	1	12781834998534.5	30_District5	1	6017528191223.9
17_District2	1	156181089016.88	3_District0	1	122561494294.58
17_District3	1	286459175916.91	3_District1	2	18208627078859.5
17_District4	2	20922289478740.1	3_District2	0	240340957506.52
17_District5	2	26532859320678.5	3_District3	1	1281223453464.44
18_District0	0	3521808556.5	3_District4	4	71112832577681.6
18_District1	2	14328913847618.3	3_District5	3	64891254385918.3
18_District2	1	8630147571.94	4_District0	1	25573824722.49
18_District3	0	93021012558.83	4_District1	2	17238235759627.9
18_District4	2	19659368358475	4_District2	0	29633450012.82
18_District5	1	13672716328157.6	4_District3	0	49572290640.61
19_District0	1	64434270618.25	4_District4	4	51890790231478.7
19_District1	2	23652212368611.5	4_District5	2	31137521929724.6
19_District2	1	219680866790.73	5_District0	0	128781182282.02
19_District3	0	346757448838.82	5_District1	2	22894699266341.7
19_District4	5	113859513555083	5_District2	1	94695484101.81
19_District5	4	95534597869554.9	5_District3	0	567433555346.88
1_District0	1	518262674109.11	5_District4	3	72230954757530.8
1_District1	3	57515558599736.1	5_District5	4	129423894047924
1_District2	1	833211902030.59	6_District0	1	220587523297.92
1_District3	1	1505646454670.73	6_District1	2	26507707681694.6
1_District4	5	78076709551623.2	6_District2	0	410060413097.7
1_District5	3	98873046645006.1	6_District3	0	365827522769.89
20_District0	0	110769617876.9	6_District4	3	37068601166034.6
20_District1	2	18197730346113.1	6_District5	3	60410529144935.3
20_District2	0	54688947926.38	7_District0	1	65702903269.61
20_District3	1	470291321930.17	7_District1	2	28027021670451.1
20_District4	3	28044909810917.7	7_District2	1	116641823753.31
20_District5	3	58730152299958.6	7_District3	0	904801723716.86
21_District0	1	147347736597.59	7_District4	3	33193370485671.2
21_District1	2	17166915036773.7	7_District5	3	59405925904447.8
21_District2	1	120541044419.74	8_District0	0	77888791330.51
21_District3	0	1297422147669.29	8_District1	1	13654929607346.6
21_District4	2	27914982366635.8	8_District2	0	87907337806.29
21_District5	3	53568470949871	8_District3	1	523196098713.99
22_District0	0	165773666324.15	8_District4	3	30291252419995.7
22_District1	1	15243631318257.1	8_District5	2	43918931439070.2
22_District2	0	324661427161.02	9_District0	0	116737451606.18
22_District3	1	474713951444.09	9_District1	1	12399153285006.3
22_District4	3	30893669654101.1	9_District2	0	136483997883.1
22_District5	3	44875375328999.7	9_District3	1	1459938832249.58
23_District0	0	40743361287.58	9_District4	3	41362999296925.9
23_District1	2	17469640737231	9_District5	2	26860109133590.2
23_District2	0	257681688762.54	C101_100t_20w	1	40805608647.36
23_District3	1	365950771984.04	C101_25t_5w	0	280350869.62
23_District4	2	38298731215190.3	C101_50t_10w	1	1573935875.42
23_District5	3	77953647815762	C102_100t_20w	1	70133116592.4
24_District0	1	151863618723.94	C102_25t_5w	0	329912424
24_District1	2	12692918197763.7	C102_50t_10w	0	5524352905.28
24_District2	0	40494552634.13	C103_100t_20w	1	158699272503.34
24_District3	1	241943658267.41	C103_25t_5w	1	327409315.8
C103_50t_10w	0	7261913029.18	R206_100t_20w	1	2083248305.76
C104_100t_20w	1	276422711492.72	R206_25t_5w	0	1559270.22
C104_25t_5w	0	287359700.52	R206_50t_10w	1	148912632.02
C104_50t_10w	0	12614845759.3	R207_100t_20w	1	602552362.26
C105_100t_20w	0	47274196432.62	R207_25t_5w	0	5341774.42
C105_25t_5w	0	288360633.06	R207_50t_10w	0	217307064.9
C105_50t_10w	0	2653093048.88	R208_100t_20w	1	402603379.84
C106_100t_20w	1	62691808364.4	R208_25t_5w	0	10348010.4
C106_25t_5w	0	283354540.84	R208_50t_10w	0	22945783
C106_50t_10w	1	2029753005.1	R209_100t_20w	1	1248330191.24
C107_100t_20w	1	70886974907.1	R209_25t_5w	0	9402282.68
C107_25t_5w	0	244305724.68	R209_50t_10w	1	18184428.68
C107_50t_10w	0	2002481692.92	R210_100t_20w	1	3672023977.68

Table D.3 – continued from previous page

Instance	Time	Objective Value	Instance	Time	Objective Value
C108_100t_20w	1	77696013979	R210_25t_5w	0	-2778914.68
C108_25t_5w	1	212766397.44	R210_50t_10w	1	69696872.28
C108_50t_10w	0	1441475837.72	R211_100t_20w	1	6230817533.52
C109_100t_20w	1	106780342246.46	R211_25t_5w	0	24364921.98
C109_25t_5w	0	285857514.98	R211_50t_10w	0	159734539.88
C109_50t_10w	0	3810168122.66	RC101_100t_20w	1	31372925621.7
C201_100t_20w	1	-3112699402.26	RC101_25t_5w	0	58017439.38
C201_25t_5w	0	-20024053.92	RC101_50t_10w	0	1481300065.14
C201_50t_10w	0	70128360.88	RC102_100t_20w	1	26501220055.86
C202_100t_20w	1	4474520475.34	RC102_25t_5w	0	48338693.22
C202_25t_5w	0	17022310.78	RC102_50t_10w	0	1267892723.54
C202_50t_10w	0	261027370.88	RC103_100t_20w	1	28873576159.68
C203_100t_20w	1	-2188612090.6	RC103_25t_5w	1	48728086.94
C203_25t_5w	0	19025148.96	RC103_50t_10w	0	1539738202.98
C203_50t_10w	0	253235287.44	RC104_100t_20w	1	39635641791.6
C204_100t_20w	1	-2869516409.62	RC104_25t_5w	0	62634465.94
C204_25t_5w	0	65582917.76	RC104_50t_10w	1	1864394369.08
C204_50t_10w	0	307777448.46	RC105_100t_20w	1	28954635863.34
C205_100t_20w	1	27114575490.06	RC105_25t_5w	0	49284339.76
C205_25t_5w	0	-20023953.88	RC105_50t_10w	0	1403815303.12
C205_50t_10w	1	-471398319.38	RC106_100t_20w	1	31351309184.32
C206_100t_20w	1	26117537853.08	RC106_25t_5w	0	62300598.44
C206_25t_5w	0	61577867.88	RC106_50t_10w	0	1596877565.7
C206_50t_10w	0	740219294.579999	RC107_100t_20w	1	37909063145.46
C207_100t_20w	1	10626970683.86	RC107_25t_5w	0	62467451.12
C207_25t_5w	0	46559137.1	RC107_50t_10w	0	1683019698.9
C207_50t_10w	1	136358501.48	RC108_100t_20w	1	43577858701.2
C208_100t_20w	1	46860791811.58	RC108_25t_5w	0	70088017.62
C208_25t_5w	0	20526829.42	RC108_50t_10w	0	1728904280.28
C208_50t_10w	0	1277850103.34	RC201_100t_20w	1	456644247.6
R101_100t_20w	1	20081267457.8	RC201_25t_5w	0	2504879.98
R101_25t_5w	0	62467584.18	RC201_50t_10w	0	-54970610.92
R101_50t_10w	0	1480434141.18	RC202_100t_20w	1	1240226021.94
R102_100t_20w	1	20032631931.82	RC202_25t_5w	0	1782126.85999999
R102_25t_5w	0	57961949.66	RC202_50t_10w	1	144585183.34
R102_50t_10w	0	1409442930.92	RC203_100t_20w	1	486368151.06
R103_100t_20w	1	25571731629.34	RC203_25t_5w	0	1504158.04
R103_25t_5w	0	49507034.04	RC203_50t_10w	0	138524275.72
R103_50t_10w	0	1470910935.92	RC204_100t_20w	1	-270192915.32
R104_100t_20w	1	37836109102.84	RC204_25t_5w	0	7177386.43999999
R104_25t_5w	0	45390782.4	RC204_50t_10w	1	11692132.36
R104_50t_10w	0	2061785102.36	RC205_100t_20w	1	2077845556.4
R105_100t_20w	1	24871913269.32	RC205_25t_5w	0	2616353.63999999
R105_25t_5w	0	61299289.8	RC205_50t_10w	0	217307534.9
R105_50t_10w	0	1417667263.1	RC206_100t_20w	1	1278052795.63999
R106_100t_20w	1	24893529568.92	RC206_25t_5w	0	1614724.18
R106_25t_5w	0	58406946.24	RC206_50t_10w	0	83548297.78
R106_50t_10w	1	1532812190.92	RC207_100t_20w	1	3777402944.77999
R107_100t_20w	1	27455025570.24	RC207_25t_5w	0	15187583.72
R107_25t_5w	0	57572567.14	RC207_50t_10w	1	24244966.86
R107_50t_10w	0	1483464446.74	RC208_100t_20w	1	10316241624.22
R108_100t_20w	1	40375989133.8	RC208_25t_5w	0	28036327.06
R108_25t_5w	0	49562611.38	RC208_50t_10w	1	216874192.84
R108_50t_10w	0	2002481236.76	hh_00_P0	2	2132214584277.35
R109_100t_20w	1	29797658783.36	ll1_00_P0	1	111689842477.59
R109_25t_5w	0	61466184.98	ll1_01_P0	1	80332194267.5
R109_50t_10w	0	1475672317.98	ll1_02_P0	1	111689842477.59
R110_100t_20w	1	33069780617.82	ll1_03_P0	1	111689842477.59
R110_25t_5w	0	65804920.66	ll1_04_P0	1	111689842477.59
R110_50t_10w	1	1544499451.28	ll1_05_P0	1	110071370621.19
R111_100t_20w	1	29805765370.18	ll1_06_P0	1	85327578093.91
R111_25t_5w	0	53511883.66	ll1_07_P0	0	108592963751.38
R111_50t_10w	0	1666137378.44	ll2_00_P0	1	11461556706.86
R112_100t_20w	0	44337120135.48	ll3_00_P0	0	6962257738.24
R112_25t_5w	0	78487214.58	test150-0-0-0-0_d0_tw0	4	2071227103475.1
R112_50t_10w	0	1942744175.54	test150-0-0-0-0_d0_tw1	4	-17864902529.2
R201_100t_20w	1	359372643	test150-0-0-0-0_d0_tw2	5	-12691662643.5
R201_25t_5w	0	-1722541.3	test150-0-0-0-0_d0_tw3	5	-11932921248.4
R201_50t_10w	0	-45880208.54	test150-0-0-0-0_d0_tw4	5	140229289294.1
R202_100t_20w	2	1194290404.7	test250-0-0-0-0_d0_tw0	9	35170169091553
R202_25t_5w	0	3116948.78	test250-0-0-0-0_d0_tw1	15	137849382242.3
R202_50t_10w	1	143285321.52	test250-0-0-0-0_d0_tw2	16	123321148400.6
R203_100t_20w	1	524193762.42	test250-0-0-0-0_d0_tw3	15	121293950820.7
R203_25t_5w	0	5842635.42	test250-0-0-0-0_d0_tw4	16	410507302095.6
R203_50t_10w	1	202589625.92	test50-0-0-0-0_d0_tw0	0	10189252978.1
R204_100t_20w	2	475557767.94	test50-0-0-0-0_d0_tw1	1	-296465074.4
R204_25t_5w	0	7010513	test50-0-0-0-0_d0_tw2	1	-324551724.1
R204_50t_10w	0	31603644.24	test50-0-0-0-0_d0_tw3	0	-327672308.4
R205_100t_20w	1	1264542313.16	test50-0-0-0-0_d0_tw4	1	-149791689.3
R205_25t_5w	0	1615002.18	BTEngineers	28	-661041733488.256
R205_50t_10w	0	82250460.88			

D.6 Results of Experiments with GH4

Table D.4: GH4 Results

Instance	Time	Objective Value	Instance	Time	Objective Value
10_District0	10	73579149872.19	24_District4	4	27578807460941

Table D.4 – continued from previous page

Instance	Time	Objective Value	Instance	Time	Objective Value
10_District1	12	8742582740243.71	24_District5	2	25800307456692.6
10_District2	2	61368525610.09	25_District0	0	2043581338.79
10_District3	1	146301147956.95	25_District1	1	6305558281369.17
10_District4	8	24328237899138	25_District2	0	4142742478.89
10_District5	5	41111683727094.2	25_District3	1	56473440051.46
11_District0	0	4356037844.01	25_District4	3	21555433539952.1
11_District1	2	6608255887344.49	25_District5	2	11538000435044.9
11_District2	0	18457519385.13	26_District0	1	241518458865.36
11_District3	1	61007292651.78	26_District1	3	45772877237830.2
11_District4	3	24910913136557.1	26_District2	1	384537188151.2
11_District5	2	16123030648188	26_District3	1	1290379002655
12_District0	0	141504604238.87	26_District4	6	116488290811905
12_District1	3	40776258487031.5	26_District5	4	88167483444268.7
12_District2	0	194867465350.16	27_District0	1	180747833715.93
12_District3	1	289344659791.84	27_District1	2	19783283701204.7
12_District4	5	68299606113628.4	27_District2	0	97662834657.88
12_District5	4	73598777477729.3	27_District3	0	172256129197.26
13_District0	0	168865119283.94	27_District4	5	36048355357399.9
13_District1	2	17128628510466	27_District5	4	41656281831128.1
13_District2	0	159231072127.29	28_District0	0	125926326083.33
13_District3	1	527501710235.23	28_District1	2	16239767493027.3
13_District4	2	30546772677029.2	28_District2	1	91586058704.99
13_District5	3	45140827999430	28_District3	1	806649183804.49
14_District0	0	40146827726.19	28_District4	4	24598363569188.8
14_District1	2	13467783817507.4	28_District5	3	34294634671496.3
14_District2	0	131035919732.48	29_District0	0	36644838335.68
14_District3	0	361447423902.35	29_District1	1	7856408345043.78
14_District4	4	35028301788981.9	29_District2	1	117630256831.99
14_District5	3	47874888280552.5	29_District3	1	286919199327.29
15_District0	1	53480666919.7	29_District4	2	9190917394665.21
15_District1	2	13282011488125.8	29_District5	4	37168164210082.8
15_District2	1	97020491372.6	2_District0	1	157617630031.37
15_District3	1	515019048175.43	2_District1	3	41093926837437.7
15_District4	4	27707465419777.2	2_District2	1	237420164797.49
15_District5	2	29560770462147	2_District3	1	886131208025.65
16_District0	1	124111535110.61	2_District4	5	65076951392060.6
16_District1	2	13946821171042.2	2_District5	4	102371743371978
16_District2	1	117994082289.6	30_District0	0	15750894375.25
16_District3	1	231546226563.76	30_District1	1	4557340049629.21
16_District4	4	31120428099885.2	30_District2	1	48192110664.85
16_District5	4	49850851504532.6	30_District3	1	151547703130.57
17_District0	0	71731794422.84	30_District4	2	6314230529921.32
17_District1	2	12088881189801.6	30_District5	1	5810068871607.39
17_District2	0	135058361618.87	3_District0	0	98810703347.8
17_District3	1	253449915597.95	3_District1	2	17931112587450.7
17_District4	4	19522415079053.5	3_District2	0	194333281733.58
17_District5	2	25900225043561.3	3_District3	1	894650902132.49
18_District0	0	2774614360.48	3_District4	5	74893769180648.8
18_District1	2	14517882908274.5	3_District5	3	64213032989375.8
18_District2	0	7306858873.38	4_District0	1	23079825892.73
18_District3	0	82811678685.22	4_District1	2	16779234384583.1
18_District4	2	20543497503197.1	4_District2	0	28173856022.96
18_District5	2	13884304730220.9	4_District3	0	41359834864.07
19_District0	0	57908349314.29	4_District4	5	49828163176811.3
19_District1	3	22726146911105.5	4_District5	3	32266174109783.7
19_District2	1	191964297514.43	5_District0	1	109880275385.19
19_District3	1	280184280055.84	5_District1	2	24388432312918.8
19_District4	5	113876873716109	5_District2	0	98814785611.31
19_District5	4	87663521737651.9	5_District3	1	367461670531.47
1_District0	1	430590936121.19	5_District4	4	73181136449680.2
1_District1	3	59476657448428.9	5_District5	5	125832972273643
1_District2	1	813278602635.29	6_District0	0	181960005448.89
1_District3	2	1122351887598.7	6_District1	3	26099439778217.9
1_District4	6	78111926829840.7	6_District2	1	289749164933.86
1_District5	5	100982821832052	6_District3	0	305431936104.79
20_District0	0	99432781249.94	6_District4	4	38503766952259.3
20_District1	2	19601334022146.7	6_District5	3	57789981077825
20_District2	1	44458138442.87	7_District0	1	56303480152.01
20_District3	1	365435739812.22	7_District1	3	27300745066741.3
20_District4	3	26358488046862.9	7_District2	1	111485778550.34
20_District5	4	50954611011141.5	7_District3	1	776501888673.67
21_District0	0	120414278382.33	7_District4	3	35776845648861.1
21_District1	2	17652202907427.1	7_District5	4	57576816725618.4
21_District2	0	115667934575.44	8_District0	0	55734117491.97
21_District3	1	973961894830.13	8_District1	2	14074384902179.4
21_District4	4	26783816491030.9	8_District2	0	78082400799.55
21_District5	3	48326719618689.2	8_District3	0	426190325963.02
22_District0	0	148175450700.46	8_District4	4	29855244503367.5
22_District1	2	15002468263354.1	8_District5	3	42953381468944.2
22_District2	1	305746328782.11	9_District0	1	107442343254.97
22_District3	0	386872327750.12	9_District1	1	12374339707973.4
22_District4	4	31295345110059.2	9_District2	0	122612311944.1
22_District5	3	44852664072610.7	9_District3	1	1161467933644.3
23_District0	1	34604221188.51	9_District4	5	40804155144146
23_District1	2	18446851538766.2	9_District5	3	26513137990724.1
23_District2	1	226177259855.41	C101_100t_20w	2	10991742137.74
23_District3	1	292513020428.46	C101_25t_5w	0	171715083.86
23_District4	4	39387173685051	C101_50t_10w	0	1776521568.06
23_District5	4	73946198034600.6	C102_100t_20w	1	26579581384.02
24_District0	0	125777297321.34	C102_25t_5w	0	51064722.8799999
24_District1	2	12334116298019.4	C102_50t_10w	0	1983003407.06
24_District2	1	31844588785.5	C103_100t_20w	2	12037418083.26
24_District3	1	174977111020.56	C103_25t_5w	0	19525564.4
C103_50t_10w	0	701260742.88	R206_100t_20w	4	435029642.38
C104_100t_20w	3	2942486164.31999	R206_25t_5w	0	-2501046.92
C104_25t_5w	0	23029997.84	R206_50t_10w	1	-26400841.76
C104_50t_10w	1	225963843.459999	R207_100t_20w	5	-253981161.74

Table D.4 – continued from previous page

Instance	Time	Objective Value	Instance	Time	Objective Value
C105_100t_20w	1	32926578587.94	R207_25t_5w	1	-3446505.1
C105_25t_5w	0	208761384.28	R207_50t_10w	1	-29430958.54
C105_50t_10w	0	1383038651.9	R208_100t_20w	5	-551200441.52
C106_100t_20w	1	11696965721.82	R208_25t_5w	0	-2055976.38
C106_25t_5w	0	97622645.1	R208_50t_10w	2	-49776178
C106_50t_10w	0	2582968265.2	R209_100t_20w	4	2023805612.76
C107_100t_20w	1	47274198496.32	R209_25t_5w	0	947578.879999999
C107_25t_5w	0	178223311.54	R209_50t_10w	0	13423441.74
C107_50t_10w	0	1979107222.62	R210_100t_20w	5	2104865022.28
C108_100t_20w	1	32950897319.16	R210_25t_5w	0	-3279615.82
C108_25t_5w	0	174719109.76	R210_50t_10w	1	99565453.5399999
C108_50t_10w	0	241547262.16	R211_100t_20w	3	7830403070.88
C109_100t_20w	2	26166175173.92	R211_25t_5w	0	1170129.94
C109_25t_5w	0	136671428.18	R211_50t_10w	1	18618600.58
C109_50t_10w	0	-327250206.46	RC101_100t_20w	1	23188568287.92
C201_100t_20w	1	-4669049241.14	RC101_25t_5w	0	49451353.64
C201_25t_5w	0	-27032513.24	RC101_50t_10w	1	1412473148.5
C201_50t_10w	1	-257123833.06	RC102_100t_20w	1	21518732405.6
C202_100t_20w	3	-3234284890.7	RC102_25t_5w	0	40217609.6
C202_25t_5w	0	-13015142.64	RC102_50t_10w	0	1334555632.24
C202_50t_10w	1	-214269133.1	RC103_100t_20w	1	18273630595.02
C203_100t_20w	5	-2699290425.82	RC103_25t_5w	0	49173343.64
C203_25t_5w	0	-26531711.88	RC103_50t_10w	1	1154046931.84
C203_50t_10w	1	175318423.22	RC104_100t_20w	1	23226396558.9
C204_100t_20w	5	-3842235579.9	RC104_25t_5w	1	49173289.36
C204_25t_5w	0	-14016174.06	RC104_50t_10w	1	1075696856.06
C204_50t_10w	1	210381362.6	RC105_100t_20w	1	20678410120.62
C205_100t_20w	2	5471558240.16	RC105_25t_5w	0	44890061.72
C205_25t_5w	0	-26531992.68	RC105_50t_10w	1	1409010160.46
C205_50t_10w	1	-401271699.16	RC106_100t_20w	1	23261521824.28
C206_100t_20w	2	-3209968089.62	RC106_25t_5w	0	57238833.18
C206_25t_5w	0	-20024005.08	RC106_50t_10w	0	1281744931.5
C206_50t_10w	1	225964531.18	RC107_100t_20w	1	19246350311.48
C207_100t_20w	3	3939524761.94	RC107_25t_5w	0	49062064.54
C207_25t_5w	0	9513283.87999999	RC107_50t_10w	1	1347109364.12
C207_50t_10w	1	724636750.9	RC108_100t_20w	1	20756768275.58
C208_100t_20w	2	-3915189482	RC108_25t_5w	0	45390814.78
C208_25t_5w	0	-31538178.98	RC108_50t_10w	0	1137597839.48
C208_50t_10w	0	342840865.36	RC201_100t_20w	1	-459329634.18
R101_100t_20w	1	19208521668.24	RC201_25t_5w	0	2226841.78
R101_25t_5w	0	53122608.32	RC201_50t_10w	0	-43281138.92
R101_50t_10w	0	1285207620.98	RC202_100t_20w	3	-478244191.4
R102_100t_20w	1	19154482285.74	RC202_25t_5w	0	-1888850.56
R102_25t_5w	0	45057049.18	RC202_50t_10w	0	-35922061.36
R102_50t_10w	0	1217679533.84	RC203_100t_20w	4	-435011993.1
R103_100t_20w	1	18422240207.14	RC203_25t_5w	0	-2277944.62
R103_25t_5w	0	49284533.14	RC203_50t_10w	1	-35922342.52
R103_50t_10w	1	954058853.18	RC204_100t_20w	4	-253978310.52
R104_100t_20w	2	19062614007.94	RC204_25t_5w	0	-2111377.74
R104_25t_5w	0	45335265.26	RC204_50t_10w	1	-40251734.54
R104_50t_10w	0	878305606.64	RC205_100t_20w	3	462051422.1
R105_100t_20w	1	17557599544	RC205_25t_5w	0	-2779131.14
R105_25t_5w	0	56849341.38	RC205_50t_10w	1	94372608.88
R105_50t_10w	0	1156643881.64	RC206_100t_20w	2	1148359696.4
R106_100t_20w	1	16744298114.48	RC206_25t_5w	0	-3167859.86
R106_25t_5w	0	48950883.28	RC206_50t_10w	1	-41982290.74
R106_50t_10w	0	739352693.46	RC207_100t_20w	2	3717960837.82
R107_100t_20w	1	15036634177.5	RC207_25t_5w	0	1003480.7
R107_25t_5w	1	44222766.02	RC207_50t_10w	0	12559459.32
R107_50t_10w	1	818135942.38	RC208_100t_20w	4	5298631393.36
R108_100t_20w	2	18451962090.24	RC208_25t_5w	1	-2278110.98
R108_25t_5w	0	49117714.96	RC208_50t_10w	0	85282092.7
R108_50t_10w	1	863155047.08	hh_00_P0	3	1124640066530.13
R109_100t_20w	1	14320603709.74	ll1_00_P0	1	80581170304.56
R109_25t_5w	1	40328946.98	ll1_01_P0	1	47480545447.34
R109_50t_10w	0	941505115.32	ll1_02_P0	1	80581170304.56
R110_100t_20w	1	16695661971.42	ll1_03_P0	1	78931585054.56
R110_25t_5w	0	61077136.84	ll1_04_P0	2	80581170304.56
R110_50t_10w	0	893023281.52	ll1_05_P0	1	78931585054.56
R111_100t_20w	1	12669682056.42	ll1_06_P0	1	55728436333.6
R111_25t_5w	0	49062035	ll1_07_P0	1	75632414554.56
R111_50t_10w	1	930683575.42	ll2_00_P0	1	5775841520.35
R112_100t_20w	1	15098779769.04	ll3_00_P0	0	5816909548.94
R112_25t_5w	0	49062021.06	test150-0-0-0-0_d0_tw0	26	47042006735.5
R112_50t_10w	0	1010332371.06	test150-0-0-0-0_d0_tw1	12	45041686643
R201_100t_20w	2	383691051.4	test150-0-0-0-0_d0_tw2	10	17244147247.7
R201_25t_5w	0	-3391171.8	test150-0-0-0-0_d0_tw3	9	49732091599
R201_50t_10w	1	-56269224.94	test150-0-0-0-0_d0_tw4	11	144436858989.3
R202_100t_20w	4	-413398450.44	test250-0-0-0-0_d0_tw0	85	434833666600.5
R202_25t_5w	0	-1443897.92	test250-0-0-0-0_d0_tw1	34	142241643737.6
R202_50t_10w	1	-48910356.92	test250-0-0-0-0_d0_tw2	31	405439315470.4
R203_100t_20w	5	-297212512.58	test250-0-0-0-0_d0_tw3	23	357800199477.5
R203_25t_5w	0	-2445283.84	test250-0-0-0-0_d0_tw4	37	635188248409.6
R203_50t_10w	1	-39387267.94	test50-0-0-0-0_d0_tw0	2	-249653877.4
R204_100t_20w	5	-426908730.04	test50-0-0-0-0_d0_tw1	1	-337034929
R204_25t_5w	0	-2723477.98	test50-0-0-0-0_d0_tw2	1	-277740696.8
R204_50t_10w	1	-43716050.8	test50-0-0-0-0_d0_tw3	0	-321431264.1
R205_100t_20w	2	1140251208.04	test50-0-0-0-0_d0_tw4	1	-227808798.4
R205_25t_5w	0	1114225.79999999	BTEngineers	14	-795238175926.579
R205_50t_10w	1	-48910479.02			

D.7 Results of Experiments with GH5

Table D.5: GH5 Results

Instance	Time	Objective Value
10_District0	49	76197329181.31
10_District1	314	8332565831754.86
10_District2	19	41566415577.57
10_District3	25	139155792160.46
10_District4	1520	21304246353721.4
10_District5	1107	37735850352798.3
11_District0	2	4030293327.26
11_District1	400	6157994611789.7
11_District2	5	16156248882.71
11_District3	22	67573409337.35
11_District4	839	22815358640054.3
11_District5	406	15870444696148.6
12_District0	9	120709766177.22
12_District1	1735	37877356021322
12_District2	18	153054550706.1
12_District3	42	249774601515.26
12_District4	3423	60511992093081.6
12_District5	1960	67378123269449.6
13_District0	8	181837406256.06
13_District1	904	16357245524195.9
13_District2	46	133430837232.11
13_District3	60	429665639491.28
13_District4	1099	30035422457655.5
13_District5	1675	42795337496140
14_District0	3	38445413734.59
14_District1	789	12437850818728.9
14_District2	11	107105283970.71
14_District3	62	274179038425.79
14_District4	2007	33558582833140.6
14_District5	903	44524141233518.7
15_District0	14	49358397440.65
15_District1	520	12323409757393.4
15_District2	15	87841832193.57
15_District3	72	464032580266.75
15_District4	2377	23674862384669.6
15_District5	813	28700034523627.8
16_District0	6	120807470457.05
16_District1	562	12709201781302.1
16_District2	15	121685669414.87
16_District3	50	210518442436.72
16_District4	1928	29236319407927.4
16_District5	1738	48522145053303.8
17_District0	4	67369227966.97
17_District1	428	12050832937058.4
17_District2	25	129187574040.27
17_District3	36	178730940805.78
17_District4	1002	19792911642330.1
17_District5	675	24540135430684.3
18_District0	0	2967361944.3
18_District1	309	13924688690304.7
18_District2	3	6610054743.83
18_District3	9	70920528499.07
18_District4	1535	19315071188565.5
18_District5	356	12846363606953.3
19_District0	4	55462969817.43
19_District1	717	22694353575366.4
19_District2	34	164098512693.08
19_District3	44	243569037679.2
19_District4	5937	108686185382264
19_District5	1768	82454118136341.8
1_District0	26	439941928923.54
1_District1	1757	57441101692321.6
1_District2	82	672084401832.35
1_District3	161	1031045184732.73
1_District4	4472	78811470071171.5
1_District5	2732	93773999626897.2
20_District0	6	99328133768.61
20_District1	626	17222315357479.6
20_District2	13	41467071363.57
20_District3	58	328595993028.43
20_District4	1028	25801083032839.4
20_District5	1613	48996376556677.3
21_District0	14	124707656273.15
21_District1	1047	16893868904739.4
21_District2	23	92151069999.22
21_District3	130	892017226174.98
21_District4	1974	25898381094219.1
21_District5	1378	45952799183238.4
22_District0	7	153203512389.48
22_District1	665	14071948171020.4
22_District2	55	226837888943.86
22_District3	43	285892462254.77
22_District4	1679	29923097625853.4
22_District5	1366	44763883711525
23_District0	3	35759011184.35
23_District1	836	17673842872150.1
23_District2	13	212716276649.99
23_District3	56	250619333021.81
24_District4	2203	24959614832931.4
24_District5	428	247409733459133.5
25_District0	1	2063904706.86
25_District1	336	6287167333080.46
25_District2	2	3529674897.16
25_District3	9	61895236430.12
25_District4	1458	20802685785650.5
25_District5	249	11017017750010.1
26_District0	26	225380467772.98
26_District1	1863	42180173025536.8
26_District2	47	347911895048.75
26_District3	190	1174205447509.03
26_District4	5174	109072516818672
26_District5	3148	76363123397244.6
27_District0	19	164846180674.81
27_District1	785	18425601260763.2
27_District2	49	69984902851.62
27_District3	38	116712931257.84
27_District4	3132	33778988955119.3
27_District5	1391	39568513214098.5
28_District0	9	118322979268.42
28_District1	587	15748059106107.7
28_District2	9	83412415745
28_District3	138	805624097687.09
28_District4	2059	22982825117023.6
28_District5	1065	34230620183401.6
29_District0	3	33680382487
29_District1	169	7485535381511.83
29_District2	28	101261828549.51
29_District3	80	242122525286.22
29_District4	737	8309110124637.32
29_District5	1181	35670467010142.6
2_District0	24	147005506881.3
2_District1	1322	41435658304765.8
2_District2	51	217829855689.62
2_District3	149	910924515484.78
2_District4	3012	58876326066126.1
2_District5	2213	95903342380824.9
30_District0	2	15793480820.97
30_District1	215	4454383868687.9
30_District2	13	36349089458.6
30_District3	30	140260859804.87
30_District4	323	6058129814232.6
30_District5	170	5492085372577.41
3_District0	15	96238044103.09
3_District1	453	18647432042214.5
3_District2	29	185092423999.03
3_District3	120	837894240881.44
3_District4	3164	67042838268903.4
3_District5	1724	61291071632688.9
4_District0	1	23017820526.83
4_District1	528	16091560843817.3
4_District2	4	26518013809.76
4_District3	13	39098434166.08
4_District4	2616	458684443223206.7
4_District5	1033	30292083288905.3
5_District0	8	106505113616.21
5_District1	885	24350894213228.5
5_District2	24	86384187127.12
5_District3	65	366709188976.7
5_District4	2935	72089499136026.7
5_District5	2753	121803986176569
6_District0	18	176185947866.7
6_District1	744	24635464198083.6
6_District2	56	276989664122.16
6_District3	49	272668316967.88
6_District4	2645	34107990031546.2
6_District5	1204	53882546018494.4
7_District0	9	53992400830.18
7_District1	882	25895048416872.9
7_District2	18	96977543097.41
7_District3	88	628351241783.24
7_District4	1767	32205982718191.7
7_District5	2101	52214576657422.1
8_District0	7	58271158883.49
8_District1	262	14440025190216.6
8_District2	16	77167528175.2
8_District3	51	367351779127.7
8_District4	1572	31102176929422.3
8_District5	1336	40455436239354.1
9_District0	5	107338834466.71
9_District1	201	12182988858340.4
9_District2	20	108517339513.82
9_District3	115	911115793606.41
9_District4	1786	38386407634635.8
9_District5	1070	25382936280280.4
C101_100t_20w	177	-5301317958.64
C101_25t_5w	1	51064664.78

Table D.5 – continued from previous page

Instance	Time	Objective Value	Instance	Time	Objective Value
23_District4	2454	34555727151866.1	C101_50t_10w	12	81815887.5
23_District5	2103	70608343796642.5	C102_100t_20w	259	-5155408825.82
24_District0	10	125755133924.82	C102_25t_5w	1	8511528.89999999
24_District1	336	11757625192420.3	C102_50t_10w	24	15586236.7599999
24_District2	6	28321020116.16	C103_100t_20w	286	-7173802275.76
24_District3	45	159315579669.34	C103_25t_5w	1	44556802.18
C103_50t_10w	28	-603857627.68	R206_100t_20w	310	-810593573.8
C104_100t_20w	338	-8122204362.62	R206_25t_5w	2	-4225347.66
C104_25t_5w	3	-33540706.86	R206_50t_10w	27	-88302653.08
C104_50t_10w	33	-806442841.52	R207_100t_20w	266	-734937958.04
C105_100t_20w	255	-6006539673.74	R207_25t_5w	3	-4392137.94
C105_25t_5w	1	49562707.54	R207_50t_10w	25	-60165906.64
C105_50t_10w	14	658405522.64	R208_100t_20w	328	-875441893.04
C106_100t_20w	263	2942485545.24	R208_25t_5w	3	-3724852
C106_25t_5w	1	62578857.5	R208_50t_10w	30	-88303079.56
C106_50t_10w	14	615551264.2	R209_100t_20w	305	-926779941.82
C107_100t_20w	306	-6906304644.96	R209_25t_5w	2	-3891629.84
C107_25t_5w	1	50564058.26	R209_50t_10w	33	-85272773
C107_50t_10w	19	179213246.38	R210_100t_20w	291	-832209555.5
C108_100t_20w	318	389094791.06	R210_25t_5w	3	-3780410.3
C108_25t_5w	1	54068510.3799999	R210_50t_10w	24	-69688758.34
C108_50t_10w	26	-436335138	R211_100t_20w	320	-870037847.14
C109_100t_20w	359	-5666086592.42	R211_25t_5w	4	-4003167.92
C109_25t_5w	2	-26031566.96	R211_50t_10w	35	-92198426.6
C109_50t_10w	27	-580482607.1	RC101_100t_20w	88	15814809991.88
C201_100t_20w	168	-5714720786.2	RC101_25t_5w	0	49785100.08
C201_25t_5w	2	-28534380.9	RC101_50t_10w	6	1140627787.76
C201_50t_10w	11	-557106740.26	RC102_100t_20w	85	15822916717.48
C202_100t_20w	196	-4547457922.28	RC102_25t_5w	1	49562717.8
C202_25t_5w	3	-25030094.06	RC102_50t_10w	8	1005137953.68
C202_50t_10w	19	-284395186.24	RC103_100t_20w	86	17527878844.96
C203_100t_20w	242	-6274036235.98	RC103_25t_5w	1	45168395.9
C203_25t_5w	3	-36043790.28	RC103_50t_10w	8	812075814.96
C203_50t_10w	28	-576585746.08	RC104_100t_20w	215	18130424572.12
C204_100t_20w	382	-7295392069.42	RC104_25t_5w	1	44167136.92
C204_25t_5w	3	-38546960.4	RC104_50t_10w	9	1328063097.1
C204_50t_10w	35	-740212682	RC105_100t_20w	92	10907978098.42
C205_100t_20w	196	-6274038330.1	RC105_25t_5w	1	40495624.02
C205_25t_5w	2	-33540709.52	RC105_50t_10w	6	1340183106.8
C205_50t_10w	15	-650607518.5	RC106_100t_20w	148	11529438265.06
C206_100t_20w	265	-7489936973.12	RC106_25t_5w	1	36212723.82
C206_25t_5w	2	-33040034.82	RC106_50t_10w	9	942370993.06
C206_50t_10w	22	-642816096.34	RC107_100t_20w	122	16638920438.66
C207_100t_20w	285	-6784715891.4	RC107_25t_5w	1	44278299.46
C207_25t_5w	2	-30537072.84	RC107_50t_10w	10	815106027.38
C207_50t_10w	25	-736316617.54	RC108_100t_20w	119	14823176358.68
C208_100t_20w	269	-8292431097.82	RC108_25t_5w	1	44222643
C208_25t_5w	2	-30537000.8	RC108_50t_10w	14	1012929814.66
C208_50t_10w	25	-767484108.8	RC201_100t_20w	255	-802485979.42
R101_100t_20w	66	14199013680.52	RC201_25t_5w	2	-1943813.54
R101_25t_5w	1	53122608.32	RC201_50t_10w	17	-74882085.56
R101_50t_10w	4	1205558731.74	RC202_100t_20w	266	-775465617.08
R102_100t_20w	79	14101742003.18	RC202_25t_5w	2	-3557855.8
R102_25t_5w	1	45057049.18	RC202_50t_10w	22	-59731503.42
R102_50t_10w	6	961417477.64	RC203_100t_20w	260	-770061049.76
R103_100t_20w	76	12469734030.74	RC203_25t_5w	3	-3446348.66
R103_25t_5w	1	47727097.78	RC203_50t_10w	25	-66656954.56
R103_50t_10w	7	748010121.8	RC204_100t_20w	334	-907863849.34
R104_100t_20w	119	16614602343.42	RC204_25t_5w	3	-3947551.9
R104_25t_5w	1	41163363.58	RC204_50t_10w	31	-95227708.42
R104_50t_10w	17	788267714.9	RC205_100t_20w	238	-756551667.86
R105_100t_20w	110	8424839701.08	RC205_25t_5w	2	-2890209.88
R105_25t_5w	0	48839515.24	RC205_50t_10w	19	-51073486.48
R105_50t_10w	8	828524787.06	RC206_100t_20w	280	-821400631.18
R106_100t_20w	133	8443754034.72	RC206_25t_5w	2	-4002513.24
R106_25t_5w	1	49173408.88	RC206_50t_10w	23	-78777648
R106_50t_10w	10	732859395.62	RC207_100t_20w	291	-894353696.9
R107_100t_20w	107	13288440448.98	RC207_25t_5w	2	-3891500.52
R107_25t_5w	1	40106459.36	RC207_50t_10w	29	-77912522.28
R107_50t_10w	10	821599264.56	RC208_100t_20w	330	-956499766.46
R108_100t_20w	243	14712393943.9	RC208_25t_5w	3	-4114144.88
R108_25t_5w	1	44389677.36	RC208_50t_10w	35	-92629680.5
R108_50t_10w	13	724635106.52	hh_00_P0	379	7645633708.15
R109_100t_20w	182	7489948051.42	ll1_00_P0	86	1432133490
R109_25t_5w	1	45112659.24	ll1_01_P0	85	1432133490
R109_50t_10w	10	806881243.16	ll1_02_P0	85	1432133490
R110_100t_20w	144	8400522008.34	ll1_03_P0	85	1432133490
R110_25t_5w	1	48338838.34	ll1_04_P0	84	1432133490
R110_50t_10w	9	931116248.04	ll1_05_P0	91	1307660206.12
R111_100t_20w	182	9257056006.9	ll1_06_P0	83	1214281948.44
R111_25t_5w	1	40829572.44	ll1_07_P0	85	1432133490
R111_50t_10w	10	804716832.2	ll2_00_P0	14	85352916.27
R112_100t_20w	193	10851236130.06	ll3_00_P0	12	47544188.25
R112_25t_5w	1	44278407.52	test150-0-0-0-0_d0_tw0	7519	10277518695
R112_50t_10w	19	597369734.16	test150-0-0-0-0_d0_tw1	6020	-27383665863.5
R201_100t_20w	204	-786275512.3	test150-0-0-0-0_d0_tw2	5230	-23452002944.5
R201_25t_5w	1	-4448153.88	test150-0-0-0-0_d0_tw3	4201	-24693580634.1
R201_50t_10w	19	-70554377.38	test150-0-0-0-0_d0_tw4	5283	-23796883882.5
R202_100t_20w	275	-699811062.22	test250-0-0-0-0_d0_tw0	54208	55072194052.8999
R202_25t_5w	2	-3836114.52	test250-0-0-0-0_d0_tw1	45467	-261508319793.6
R202_50t_10w	23	-50641827.9	test250-0-0-0-0_d0_tw2	38968	-238195559324.4
R203_100t_20w	263	-780871434.02	test250-0-0-0-0_d0_tw3	34592	-182447655507.1
R203_25t_5w	2	-3836165.9	test250-0-0-0-0_d0_tw4	41802	-159472761238.6
R203_50t_10w	24	-66658297.56	test50-0-0-0-0_d0_tw0	130	-343277063
R204_100t_20w	337	-956501871.56	test50-0-0-0-0_d0_tw1	80	-396330084.6
R204_25t_5w	3	-4002989.46	test50-0-0-0-0_d0_tw2	60	-386967451.4
R204_50t_10w	30	-89601106.16	test50-0-0-0-0_d0_tw3	50	-386967676.6

Table D.5 – continued from previous page

Instance	Time	Objective Value	Instance	Time	Objective Value
R205_100t_20w	254	-837613543.32	test50-0-0-0-0_d0_tw4	90	-346397693.9
R205_25t_5w	3	-4448041.9	BTEngineers	8824	-805178654215.936
R205_50t_10w	24	-88735610.36			

D.8 Results of Experiments with GH5

Table D.6: GH5 results incrementing the *maxBranching* parameter to values 20, 40 and 50. Time is also given for each instance in milliseconds

Instance	T(20)	Objective(20)	T(40)	Objective(40)	T(50)	Objective(50)
10_District0	60	76197329181.31	54	76197329181.31	49	76197329181.31
10_District1	660	8218857439424.1	688	8218857439409.06	684	8218857439409.06
10_District2	20	41566415577.57	22	41566415577.57	22	41566415577.57
10_District3	30	139155792160.46	25	139155792160.46	26	139155792160.46
10_District4	3050	21291228411410.1	4622	22416897586296.1	5575	22035548436752.1
10_District5	2330	37249297856014.8	4276	37711823069094	4480	37688796922293
11_District0	10	4030293327.26	2	4030293327.26	2	4030293327.26
11_District1	750	6690150999911.55	1115	6362570546289.82	1103	6362570546289.82
11_District2	10	16156248882.71	6	16156248882.71	6	16156248882.71
11_District3	20	67389097303.84	28	67389097303.84	29	67389097303.84
11_District4	1400	22460659385239.4	1875	23106951893758.8	1825	23112754743284.8
11_District5	900	15860944782249.3	743	15643005576898.3	753	15643005576898.3
12_District0	10	120709766177.22	10	120709766177.22	10	120709766177.22
12_District1	4161	38784501989049.7	7697	38789221791332.6	8081	38785445949807.6
12_District2	20	153054550706.1	19	153054550706.1	19	153054550706.1
12_District3	50	239836499336.7	50	239836499336.7	51	239836499336.7
12_District4	6501	62031037751001.6	14577	59742449173420.8	16252	60514664116154.3
12_District5	2880	65945373929847	3129	65945373929847	3157	65945373929847
13_District0	10	181837406256.06	8	181837406256.06	7	181837406256.06
13_District1	2551	14674175609787.1	6518	16119941326279.3	6874	16119941326279.3
13_District2	60	126904696641.59	67	126837068235.01	64	126837068235.01
13_District3	60	429665639491.28	64	429665639491.28	64	429665639491.28
13_District4	2160	30038955101714.3	3583	30033656135618.7	4052	30033656135625.8
13_District5	2170	42764648834839.3	2384	42764648834862.9	2485	42764648834839.3
14_District0	0	38445413734.59	3	38445413734.59	3	38445413734.59
14_District1	1790	12434388027267.4	2369	12824694048872.9	2371	12824694048872.9
14_District2	30	100681954876.7	25	100681954876.7	24	100681954876.7
14_District3	90	262406322982.98	124	262406322982.98	123	262406322982.98
14_District4	4741	31546738861338.4	10027	31560980324208.6	11750	31560980324208.6
14_District5	1011	44524141233501.8	1094	44524141233501.8	1107	44524141233501.8
15_District0	10	49358397440.65	13	49358397440.65	13	49358397440.65
15_District1	740	12322474536374.6	1123	12317798430422.9	1135	12317798430422.9
15_District2	20	88202791010.04	13	88202791010.04	13	88202791010.04
15_District3	90	463823619193.16	88	463823619193.16	91	463823619193.16
15_District4	5890	23666725574721.6	12081	22834329911998.4	12632	22847348807714.7
15_District5	3211	29066199377964.7	1790	29086910060862	1798	29086910060862
16_District0	0	120807470457.05	6	120807470457.05	6	120807470457.05
16_District1	1070	12316160134202.9	1353	12717924637363.6	1454	12717924637363.6
16_District2	10	121685669414.87	16	121685669414.87	16	121685669414.87
16_District3	60	210518442436.72	60	210518442436.72	62	210518442436.72
16_District4	4821	28779433928059.5	10094	28327769996973.5	10193	28779433928105.7
16_District5	2470	50336936612873	3380	49733482433960.8	3512	49733482433960.8
17_District0	0	67369227966.97	4	67369227966.97	4	67369227966.97
17_District1	670	12405157292750.4	774	12232037741729.1	803	12232988948041.1
17_District2	30	129187574040.27	27	129187574040.27	27	129187574040.27
17_District3	40	178730940805.78	40	178730940805.78	40	178730940805.78
17_District4	1810	19092974442252.1	2566	19775413212128.2	2649	19775413212128.2
17_District5	1020	24205298553968.2	996	23886759047749.6	1006	23886759047749.6
18_District0	0	2967361944.3	0	2967361944.3	0	2967361944.3
18_District1	300	14313538412508.2	311	14127545242174.4	321	14127545242174.4
18_District2	0	6610054743.83	3	6610054743.83	4	6610054743.83
18_District3	10	70920528499.07	9	70920528499.07	9	70920528499.07
18_District4	3480	18728466766745	7846	18718722507656.5	8391	19012349527135.2
18_District5	600	12844784588836.5	726	12641617614178.6	741	12643196632098.1
19_District0	10	55462969817.43	5	55462969817.43	4	55462969817.43
19_District1	1120	22695030028950.2	1168	22401449014876.9	1206	22401449014876.9
19_District2	50	163837387274.93	58	163762780181.55	58	163762780181.55
19_District3	50	243657801812.07	56	243657801812.07	56	243657801812.07
19_District4	13961	106192108824987	36938	110989300160535	8619	112281667750350
19_District5	5200	83157516755236.4	15961	82326503699401.3	19878	78807991381031.8
1_District0	30	439941928923.54	26	439941928923.54	26	439941928923.54
1_District1	4790	58041560615784.5	3084	58075186316894.3	3162	58075186316894.3
1_District2	100	672001346398.58	111	672001346398.58	113	672001346398.58
1_District3	200	1031045184744.94	203	1031045184732.73	203	1031045184732.73
1_District4	9470	74782293092736.1	2896116	74707056177177.4	27718	73732177837593.4
1_District5	4280	91806570717137.9	14224	92564056264174.5	15373	92564056264174.5
20_District0	0	99328133768.61	6	99328133768.61	6	99328133768.61
20_District1	960	17693265129798.2	1302	18159592081819.7	1273	18159592081819.7
20_District2	10	41484065747.74	14	41484065747.74	14	41484065747.74
20_District3	70	305213925793.81	71	305213925793.81	74	305213925793.81
20_District4	1750	24960379220572.1	2267	25384073884973.4	4657	25027234394706.9
20_District5	2590	50193825670337.9	3047	48965990159239.8	3098	48965990159239.8
21_District0	20	124707656273.15	15	124707656273.15	15	124707656273.15
21_District1	1020	17340723811259.4	1153	17340723811225.2	1170	17340723811225.2
21_District2	30	92151069999.22	30	92151069999.22	32	92151069999.22
21_District3	150	894018445649.06	150	894018445649.06	149	894018445649.06

Table D.6 – continued from previous page

Instance	T(20)	Objective(20)	T(40)	Objective(40)	T(50)	Objective(50)
21_District4	5100	25489098854436.4	11830	24304758516183	12650	25468151338417.6
21_District5	1970	45930297567746.2	2281	45930297567747.9	2318	45930297567747.9
22_District0	10	153203512389.48	7	153203512389.48	7	153203512389.48
22_District1	1300	14101603952448.2	1651	14081460402643	1659	14081460402643
22_District2	70	226981185052.56	75	226981185052.56	64	226981185052.56
22_District3	40	285892462254.77	42	285892462254.77	43	285892462254.77
22_District4	3250	29074924558505.3	5066	29437811612757	5352	29492977340536
22_District5	1290	44259074452298.5	1638	43732586267201.7	1689	43732586267200.1
23_District0	0	35759011184.35	2	35759011184.35	2	35759011184.35
23_District1	1760	17656086164467.9	2685	17411043600471.4	2743	17411043600471.4
23_District2	10	212716276649.99	14	212716276649.99	13	212716276649.99
23_District3	60	250619333021.81	57	250619333021.81	58	250619333021.81
23_District4	3690	36066134523281.9	9379	35554411807447.9	11155	35549253312349.6
23_District5	3900	71437022513346.6	5476	71437022513346.6	5579	71437022513346.6
24_District0	10	125755133924.82	9	125755133924.82	9	125755133924.82
24_District1	510	11389191027794.5	606	11389191027676.9	613	11389191027680.7
24_District2	0	28321020116.16	6	28321020116.16	6	28321020116.16
24_District3	50	159315579521.22	49	159315579521.22	49	159315579521.22
24_District4	5630	24542616265297	12206	24517137738123.8	15306	25782571270930
24_District5	540	24738655441667.6	550	24738655441667.6	554	24738655441667.6
25_District0	0	2063904706.86	1	2063904706.86	1	2063904706.86
25_District1	710	6071232244353.83	869	6180626672478.48	861	6180626672478.48
25_District2	0	3529674897.16	1	3529674897.16	1	3529674897.16
25_District3	10	61895236430.12	9	61895236430.12	9	61895236430.12
25_District4	3580	20816398351963.9	6594	21142613103295.2	7028	20503174454937
25_District5	790	11011719621667.4	310	11014368685470.2	313	11014368685470.2
26_District0	30	225380467772.98	30	225380467772.98	30	225380467772.98
26_District1	4220	43684739705319.4	2235	45172896658365.8	2244	45172896658365.8
26_District2	50	347911895048.75	50	347911895048.75	51	347911895048.75
26_District3	310	1171533328289.04	368	1171533328289.04	371	1171533328289.04
26_District4	16040	105665970976886	27643	106745235908029	29378	105580668249051
26_District5	6000	78133548666857	7967	78206744543738.2	8608	78206744543738.2
27_District0	20	164846180674.81	19	164846180674.81	20	164846180674.81
27_District1	1840	18940261974778	3021	18425601260045.1	3063	18425601260045.1
27_District2	70	61192189969.07	71	61192189969.07	73	61192189969.07
27_District3	30	116712931257.84	38	116712931257.84	38	116712931257.84
27_District4	4560	33841394054854.4	6026	34358464881058.4	6216	34363417666721.2
27_District5	1970	40606510956597	2253	40597681108258.2	2273	40597681108258.2
28_District0	10	118322979268.42	9	118322979268.42	9	118322979268.42
28_District1	780	15288488523125.3	2545	15275097772158.7	2544	15275097772158.7
28_District2	10	83412415745	11	83412415745	11	83412415745
28_District3	150	805624097687.09	159	805624097687.09	159	805624097687.09
28_District4	4990	23356070207999.9	2834	24533901196890.1	3292	24924654586133.5
28_District5	1540	33797846183526.1	1695	33797846183526.1	1728	33791534896023.1
29_District0	0	33680382487	2	33680382487	3	33680382487
29_District1	220	7485535381511.83	243	7484826255380.25	253	7484826255380.25
29_District2	40	101233312043.57	44	101233312043.57	45	101233312043.57
29_District3	130	232601979076.29	125	232601979076.29	123	232601979076.29
29_District4	1320	8324751001612.9	2253	8479891590329.35	2269	8479891590329.35
29_District5	1310	36609335942077.2	1444	36128200717047.8	1474	36128200717047.8
2_District0	30	147005506881.3	26	147005506881.3	26	147005506881.3
2_District1	3060	39540340925649.8	5939	41887471336771	5913	41887471336771
2_District2	50	226411848749	51	226411848749	51	226411848749
2_District3	180	858787147589.81	177	858787147589.81	179	858787147589.81
2_District4	6430	59605501704104.2	11777	60387325402546.8	13016	60387325402546.8
2_District5	2719	95903342380824.9	3137	95903342380824.9	3240	95903342380824.9
30_District0	10	15793480820.97	2	15793480820.97	2	15793480820.97
30_District1	440	4366718210337.8	628	4288991390631.01	619	4288991390631.01
30_District2	20	36349089458.6	15	36349089458.6	15	36349089458.6
30_District3	30	140260859804.87	28	140260859804.87	29	140260859804.87
30_District4	730	6296033900094.92	1815	6033193692111.07	1831	6033193692111.07
30_District5	200	5499634837860.3	229	5491783393822.48	229	5491783393822.48
3_District0	10	96238044103.09	16	96238044103.09	16	96238044103.09
3_District1	1100	18391857798945	1906	18644467143297.4	1928	18644467143297.4
3_District2	40	184423936337.49	46	184423936337.49	42	184423936337.49
3_District3	150	838602434511.32	147	838602434511.32	147	838602434511.32
3_District4	6440	67073753531539.1	8993	67096939978154.6	9944	67062933189446.7
3_District5	3350	60566955554266.6	4408	605682304065557	4445	605682304065557
4_District0	0	23017820526.83	1	23017820526.83	2	23017820526.83
4_District1	1210	16733941829482.2	2244	16524601852384.6	2241	16524601852384.6
4_District2	0	26518013809.76	3	26518013809.76	4	26518013809.76
4_District3	20	39098434166.08	14	39098434166.08	14	39098434166.08
4_District4	8120	44497725128783	14587	44517970313461.3	16916	44535833711225.8
4_District5	1370	30671942398887.4	1437	30671942398887.4	1438	30671942398887.4
5_District0	10	106505113616.21	9	106505113616.21	8	106505113616.21
5_District1	2980	23401109473559.2	3459	23711329992067.2	3473	23711329992067.2
5_District2	20	86384187127.12	24	86384187127.12	24	86384187127.12
5_District3	60	366270241695.49	67	366270241695.49	67	366270241695.49
5_District4	3270	72163799058376.3	17633	73686947454562.1	21861	73686947454562.1
5_District5	3940	121928573278019	4299	121928573278019	4314	121928573278019
6_District0	20	176185947866.7	19	176185947866.7	27	176185947866.7
6_District1	990	24656791625774.8	1152	24652221462664	1186	24652221462664
6_District2	60	276989664122.16	57	276989664122.16	57	276989664122.16
6_District3	60	272026860685.19	55	272026860685.19	56	272026860685.19
6_District4	6410	34599408667054.7	13943	32674685677626.2	15843	32671893526336.1
6_District5	1800	52577668995018.6	1861	52577668995018.6	1886	52577668995018.6
7_District0	10	53992400830.18	10	53992400830.18	9	53992400830.18
7_District1	1567	25530776489059.8	1893	25530776489096.9	1922	25530776489086.5
7_District2	20	96812988604.09	20	96812988604.09	20	96812988604.09
7_District3	110	628007504612.66	117	627921570619.92	118	627921570619.92
7_District4	4901	32603036789102.8	6299	31729342919882.8	7092	30888008083772.9
7_District5	3751	53490232800098.6	4768	53499673363687.4	4805	53499673363687.4
8_District0	10	58271158883.49	7	58271158883.49	6	58271158883.49
8_District1	270	14083437894143.3	2291	13854598374729.3	2270	13854598374729.3
8_District2	20	73965473501.2	17	73965473501.2	16	73965473501.2
8_District3	60	367102727288.89	67	367102727288.89	66	367102727288.89
8_District4	3501	30204218209504.6	6884	29816748603227.9	6661	29831812216156.8
8_District5	2011	40450402089088.4	2282	40450402089088.4	2259	40450402089088.4

Table D.6 – continued from previous page

Instance	T(20)	Objective(20)	T(40)	Objective(40)	T(50)	Objective(50)
9_District0	10	107338834466.71	6	107338834466.71	5	107338834466.71
9_District1	260	12178217016824.2	304	12178217016824.2	314	12178217016824.2
9_District2	20	108377785409.3	21	108377785409.3	22	108377785409.3
9_District3	130	941624656698.36	139	941624656698.36	139	941624656698.36
9_District4	5661	39581950182572.1	10923	39520093387201.2	12626	37164202673658.1
9_District5	1610	25386852433132.7	1904	25385285972093.7	1922	25385285972093.7
C101_100t_20w	170	-5301317958.64	165	-5301317958.64	183	-5301317958.64
C101_25t_5w	0	51064664.78	1	51064664.78	0	51064664.78
C101_50t_10w	10	81815887.5	12	81815887.5	11	81815887.5
C102_100t_20w	400	-4912229242.02	444	-4912229242.02	439	-4912229242.02
C102_25t_5w	0	8511528.89999999	1	8511528.89999999	2	8511528.89999999
C102_50t_10w	20	15586349.0599999	28	15586349.0599999	28	15586349.0599999
C103_100t_20w	540	-7222438475.86	678	-7173802345.74	676	-7173802345.74
C103_25t_5w	0	44556802.18	2	44556802.18	2	44556802.18
C103_50t_10w	30	-564899047.04	37	-564899047.04	37	-564899047.04
C104_100t_20w	871	-8608564628.28	1808	-8681519205.24	1910	-8681519205.24
C104_25t_5w	0	-33540706.86	2	-33540706.86	2	-33540706.86
C104_50t_10w	60	-783067878.78	71	-783067878.78	72	-783067878.78
C105_100t_20w	280	-7125167921.22	286	-7125167921.22	290	-7125167921.22
C105_25t_5w	0	49562707.54	1	49562707.54	2	49562707.54
C105_50t_10w	10	658405522.64	15	658405522.64	15	658405522.64
C106_100t_20w	310	-5349954009.76	320	-5349954009.76	320	-5349954009.76
C106_25t_5w	0	62578857.5	1	62578857.5	1	62578857.5
C106_50t_10w	20	615551264.2	24	615551264.2	14	615551264.2
C107_100t_20w	370	-6979258964.62	376	-6979258964.62	376	-6979258964.62
C107_25t_5w	0	50564058.26	1	50564058.26	2	50564058.26
C107_50t_10w	20	179213246.38	19	179213246.38	19	179213246.38
C108_100t_20w	450	-6492899831.1	462	-6492899831.1	451	-6492899831.1
C108_25t_5w	10	54068510.3799999	1	54068510.3799999	2	54068510.3799999
C108_50t_10w	20	-436335138	21	-436335138	21	-436335138
C109_100t_20w	620	-6614488230.82	650	-6614488230.82	636	-6614488230.82
C109_25t_5w	0	-26031566.96	1	-26031566.96	1	-26031566.96
C109_50t_10w	30	-580482607.1	28	-580482607.1	30	-580482607.1
C201_100t_20w	160	-5714720786.2	175	-5714720786.2	179	-5714720786.2
C201_25t_5w	0	-28534380.9	1	-28534380.9	1	-28534380.9
C201_50t_10w	10	-557106740.26	10	-557106740.26	11	-557106740.26
C202_100t_20w	200	-5811995780.2	307	-5447225594.96	341	-6055175633.16
C202_25t_5w	0	-25030029.44	4	-27533187.48	3	-27533187.48
C202_50t_10w	30	-284395355.62	41	-284395408.1	52	-292187247.64
C203_100t_20w	470	-6687442461.34	886	-6930623788.88	1317	-6638806995.04
C203_25t_5w	0	-37045193.76	4	-37045195.22	4	-37045195.22
C203_50t_10w	50	-588273655.86	83	-599961178.78	96	-599961229.4
C204_100t_20w	800	-7854704998.88	1325	-8414020486.86	3111	-8657198950.24
C204_25t_5w	0	-38546960.4	4	-38546960.4	4	-38546960.4
C204_50t_10w	60	-802546635.64	124	-860985069.72	138	-880464508.52
C205_100t_20w	220	-6322674065.22	216	-6322674065.22	215	-6322674065.22
C205_25t_5w	0	-33540709.52	2	-33540709.52	2	-33540709.52
C205_50t_10w	20	-650607518.5	31	-650607518.5	15	-650607518.5
C206_100t_20w	360	-7733117210.96	370	-7733117210.96	370	-7733117210.96
C206_25t_5w	0	-33040034.82	2	-33040034.82	2	-33040034.82
C206_50t_10w	30	-670087201.58	13	-670087201.58	24	-670087201.58
C207_100t_20w	480	-7222438653.24	358	-6809033308.7	349	-6809033308.7
C207_25t_5w	0	-30537072.84	2	-30537072.84	2	-30537072.84
C207_50t_10w	30	-724629447.8	37	-712941217.36	36	-712941217.36
C208_100t_20w	400	-8608564830.12	427	-8657201177.38	427	-8657201177.38
C208_25t_5w	0	-30537000.8	1	-30537000.8	2	-30537000.8
C208_50t_10w	20	-767484108.8	28	-767484108.8	27	-767484108.8
R101_100t_20w	60	14199013680.52	66	14199013680.52	66	14199013680.52
R101_25t_5w	0	53122608.32	0	53122608.32	0	53122608.32
R101_50t_10w	10	1205558731.74	4	1205558731.74	4	1205558731.74
R102_100t_20w	70	14101742003.18	78	14101742003.18	78	14101742003.18
R102_25t_5w	0	45057049.18	0	45057049.18	0	45057049.18
R102_50t_10w	10	961417477.64	6	961417477.64	6	961417477.64
R103_100t_20w	80	12469734030.74	77	12469734030.74	76	12469734030.74
R103_25t_5w	0	47727097.78	1	47727097.78	1	47727097.78
R103_50t_10w	10	748010121.8	7	748010121.8	7	748010121.8
R104_100t_20w	320	15579736366.16	451	14023383931.96	445	14023383931.96
R104_25t_5w	0	41163363.58	1	41163363.58	1	41163363.58
R104_50t_10w	20	795626623.76	18	795626623.76	20	795626623.76
R105_100t_20w	110	8424839701.08	111	8424839701.08	112	8424839701.08
R105_25t_5w	0	48839515.24	1	48839515.24	1	48839515.24
R105_50t_10w	0	828524787.06	7	828524787.06	7	828524787.06
R106_100t_20w	120	8443754034.72	133	8443754034.72	133	8443754034.72
R106_25t_5w	0	49173408.88	1	49173408.88	1	49173408.88
R106_50t_10w	20	732859395.62	11	732859395.62	12	732859395.62
R107_100t_20w	110	11634816307.34	115	11634816307.34	114	11634816307.34
R107_25t_5w	0	40106459.36	1	40106459.36	1	40106459.36
R107_50t_10w	10	821599264.56	8	821599264.56	8	821599264.56
R108_100t_20w	320	14715096346.74	443	12369760088.08	441	12369760088.08
R108_25t_5w	0	44389677.36	1	44389677.36	0	44389677.36
R108_50t_10w	10	724635106.52	14	724635106.52	14	724635106.52
R109_100t_20w	200	6636116061.12	200	6636116061.12	200	6636116061.12
R109_25t_5w	0	45112659.24	1	45112659.24	1	45112659.24
R109_50t_10w	10	806881243.16	11	806881243.16	10	806881243.16
R110_100t_20w	150	7622345982.82	155	7622345982.82	156	7622345982.82
R110_25t_5w	0	48338838.34	1	48338838.34	0	48338838.34
R110_50t_10w	10	931116248.04	12	931116248.04	9	931116248.04
R111_100t_20w	190	8473476077.26	201	8473476077.26	200	8473476077.26
R111_25t_5w	0	40829572.44	1	40829572.44	1	40829572.44
R111_50t_10w	10	804716832.2	9	804716832.2	9	804716832.2
R112_100t_20w	290	10910679997.96	315	11618604066.8	315	11618604066.8
R112_25t_5w	0	44278407.52	1	44278407.52	1	44278407.52
R112_50t_10w	10	597369734.16	18	597369734.16	17	597369734.16
R201_100t_20w	230	-794381502.58	225	-794381502.58	235	-794381502.58
R201_25t_5w	0	-4448153.88	2	-4448153.88	1	-4448153.88
R201_50t_10w	10	-70554377.38	18	-70554377.38	20	-70554377.38
R202_100t_20w	361	-778169509.08	428	-778169072.06	466	-743043085.54
R202_25t_5w	0	-3947227.18	3	-3947227.18	3	-3947227.18

Table D.6 – continued from previous page

Instance	T(20)	Objective(20)	T(40)	Objective(40)	T(50)	Objective(50)
R202_50t_10w	30	-87436435.78	43	-78346243.1	45	-78346224.06
R203_100t_20w	470	-807891511.9	933	-891653653.54	1153	-886249252.04
R203_25t_5w	0	-4114190.48	4	-4114172.12	4	-4114172.12
R203_50t_10w	40	-64927278.02	68	-69689050.08	81	-70554528.3
R204_100t_20w	700	-934885986.92	1655	-1007840112.08	2390	-1002435489.24
R204_25t_5w	10	-3836176.22	6	-3836176.22	6	-3836176.22
R204_50t_10w	50	-87437532.82	83	-95229116.26	82	-95229116.26
R205_100t_20w	560	-910567724.8	615	-907865771.62	617	-907865771.62
R205_25t_5w	0	-4448041.9	3	-4448041.9	3	-4448041.9
R205_50t_10w	30	-88735610.36	32	-88735610.36	32	-88735610.36
R206_100t_20w	650	-932183729.82	840	-964607487.18	887	-959203558.62
R206_25t_5w	0	-4225347.66	3	-4225347.66	3	-4225347.66
R206_50t_10w	40	-80943919.4	27	-79212126.44	46	-79212126.44
R207_100t_20w	620	-967309992.24	1109	-1018647805.1	1279	-1005137727.98
R207_25t_5w	0	-4392198.74	4	-4392198.74	4	-4392198.74
R207_50t_10w	40	-61464268.7	65	-65360484.06	76	-66226236.7
R208_100t_20w	710	-951097400.26	1751	-1042965700.1	1777	-1007839487.38
R208_25t_5w	0	-3836155.76	5	-3836176.22	5	-3836176.22
R208_50t_10w	60	-92198895.8	87	-90467308.72	85	-90467308.72
R209_100t_20w	630	-959203840.68	1154	-1013243421.1	1356	-924077775.82
R209_25t_5w	0	-3891629.84	2	-3891629.84	2	-3891629.84
R209_50t_10w	30	-88735190.58	44	-88735190.58	41	-88735190.58
R210_100t_20w	520	-978117773.22	605	-924077075.12	621	-929481081.68
R210_25t_5w	0	-4336664.4	3	-4336664.4	6	-4336664.4
R210_50t_10w	30	-71853394.54	46	-70554826.58	45	-70554826.58
R211_100t_20w	660	-937588195.2	1103	-942991911.98	1104	-942991911.98
R211_25t_5w	0	-4003167.92	3	-4003167.92	3	-4003167.92
R211_50t_10w	50	-88735718.58	56	-88735718.58	55	-88735718.58
RC101_100t_20w	90	15814809991.88	88	15814809991.88	88	15814809991.88
RC101_25t_5w	0	49785100.08	0	49785100.08	0	49785100.08
RC101_50t_10w	0	1140627787.76	5	1140627787.76	5	1140627787.76
RC102_100t_20w	80	15822916717.48	85	15822916717.48	85	15822916717.48
RC102_25t_5w	0	49562717.8	0	49562717.8	1	49562717.8
RC102_50t_10w	10	1005137953.68	7	1005137953.68	8	1005137953.68
RC103_100t_20w	90	17527878844.96	86	17527878844.96	85	17527878844.96
RC103_25t_5w	0	45168395.9	1	45168395.9	0	45168395.9
RC103_50t_10w	10	812075814.96	8	812075814.96	8	812075814.96
RC104_100t_20w	90	19862406746.3	81	19862406746.3	89	19862406746.3
RC104_25t_5w	0	44167136.92	0	44167136.92	1	44167136.92
RC104_50t_10w	10	1318539952.42	13	1318539952.42	14	1318539952.42
RC105_100t_20w	90	10907978098.42	93	10907978098.42	92	10907978098.42
RC105_25t_5w	0	40495624.02	1	40495624.02	1	40495624.02
RC105_50t_10w	10	1340183106.8	6	1340183106.8	6	1340183106.8
RC106_100t_20w	150	12350846197.84	152	12350846197.84	152	12350846197.84
RC106_25t_5w	0	36212723.82	1	36212723.82	1	36212723.82
RC106_50t_10w	10	942370993.06	9	942370993.06	8	942370993.06
RC107_100t_20w	130	16638920438.66	140	16638920438.66	128	16638920438.66
RC107_25t_5w	0	44278299.46	0	44278299.46	1	44278299.46
RC107_50t_10w	10	815106027.38	10	815106027.38	12	815106027.38
RC108_100t_20w	120	15674306402.32	120	15674306402.32	123	15674306402.32
RC108_25t_5w	0	44222643	1	44222643	1	44222643
RC108_50t_10w	10	1012929814.66	14	1012929814.66	13	1012929814.66
RC201_100t_20w	290	-826803732.5	292	-826803732.5	292	-826803732.5
RC201_25t_5w	10	-1943813.54	1	-1943813.54	1	-1943813.54
RC201_50t_10w	20	-74882085.56	21	-74882085.56	17	-74882085.56
RC202_100t_20w	410	-767358429.54	464	-848419159.52	538	-848418163.6
RC202_25t_5w	0	-3557855.8	3	-3557855.8	3	-3557855.8
RC202_50t_10w	30	-67523371.86	35	-76180581.06	36	-76180581.06
RC203_100t_20w	450	-783571904.54	897	-864631843.02	1195	-915968950.18
RC203_25t_5w	10	-3669285.62	4	-3669285.62	3	-3669285.62
RC203_50t_10w	40	-70120280.08	60	-70986078.54	62	-70986078.54
RC204_100t_20w	720	-961903747.32	1691	-1018645842.94	1993	-1032155821.9
RC204_25t_5w	10	-3836301.9	4	-3836301.9	3	-3836301.9
RC204_50t_10w	50	-98691115.84	77	-96526125.94	73	-96526125.94
RC205_100t_20w	350	-805187877.12	390	-856525543.8	383	-856525543.8
RC205_25t_5w	10	-3279730.22	2	-3279730.22	2	-3279730.22
RC205_50t_10w	20	-58865195.74	26	-58865195.74	26	-58865195.74
RC206_100t_20w	580	-905161565.1	488	-902460332.16	487	-902460332.16
RC206_25t_5w	0	-4002513.24	6	-4002513.24	2	-4002513.24
RC206_50t_10w	30	-80508818.28	26	-80508818.28	25	-80508818.28
RC207_100t_20w	620	-907864881.66	1170	-891650951.2	1243	-891650951.2
RC207_25t_5w	10	-3891500.52	2	-3891500.52	2	-3891500.52
RC207_50t_10w	50	-81375438.26	48	-78345400.66	51	-78345400.66
RC208_100t_20w	720	-978116228.2	1275	-972711902.2	1228	-970010666.38
RC208_25t_5w	10	-4114144.88	4	-4114144.88	3	-4114144.88
RC208_50t_10w	50	-88301331.26	52	-88301321.2	51	-88301321.2
hh_00_P0	540	6663651008.29	603	7224779745.15	606	7224779745.15
ll1_00_P0	90	1338732826.5	93	1338732826.5	93	1338732826.5
ll1_01_P0	90	1338732826.5	93	1338732826.5	92	1338732826.5
ll1_02_P0	90	1338732826.5	94	1338732826.5	95	1338732826.5
ll1_03_P0	90	1338732826.5	97	1338732826.5	92	1338732826.5
ll1_04_P0	90	1338732826.5	83	1338732826.5	93	1338732826.5
ll1_05_P0	100	1338732826.5	106	1338732826.5	105	1338732826.5
ll1_06_P0	80	1245408125.36	89	1245408125.36	89	1245408125.36
ll1_07_P0	90	1338732826.5	93	1338732826.5	93	1338732826.5
ll2_00_P0	20	85352916.27	14	85352916.27	14	85352916.27
ll3_00_P0	20	47544188.25	14	47544188.25	14	47544188.25
test150-0-0-0-0_d0_tw0	19550	-19865223626.7	47847	-28004454190.5	61648	-28349336446.9
test150-0-0-0-0_d0_tw1	13830	-26969806509.5	25729	-30211702649.1	29881	-30832491493.7
test150-0-0-0-0_d0_tw2	10120	-27728547305.4	18937	-28832172176.8	20353	-27107759329.8
test150-0-0-0-0_d0_tw3	6670	-27383664735	7859	-26417993157.2	7869	-26417993157.2
test150-0-0-0-0_d0_tw4	10300	-26073109561.3	14382	-26142085667	14994	-26142085667
test250-0-0-0-0_d0_tw0	164208	2986736345339.9	430736	251710248339.1	545000	23650649218.5
test250-0-0-0-0_d0_tw1	109491	-276036560388.2	5138199	-289889069536.6	266964	-292254131472
test250-0-0-0-0_d0_tw2	80722	-254413132190.1	143139	-275698693917.1	153535	-271982167429.3
test250-0-0-0-0_d0_tw3	50759	-266914175565	55232	-257791791244	55129	-257791791244
test250-0-0-0-0_d0_tw4	98189	-177041797000.3	196739	-219612925007.3	211644	-231776103927.7
test50-0-0-0-0_d0_tw0	230	-365122278.5	367	-346397868.1	380	-346397868.1

Table D.6 – continued from previous page

Instance	T(20)	Objective(20)	T(40)	Objective(40)	T(50)	Objective(50)
test50-0-0-0-0_d0_tw1	100	-408812648.4	97	-408812648.4	96	-408812648.4
test50-0-0-0-0_d0_tw2	80	-408812648.4	82	-408812648.4	81	-408812648.4
test50-0-0-0-0_d0_tw3	60	-408812648.4	69	-408812648.4	68	-408812648.4
test50-0-0-0-0_d0_tw4	140	-346397693.9	159	-346397693.9	161	-346397693.9
BTEngineers	13114	-795238177220.579	16107	-795238177264.579	16108	-795238177264.579

Appendix E

Results Tabu Search Configurations

E.1 Tabu Search Results: Config. 1

Table E.1: Tabu Search experiments results with parameter configuration 1

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
10_District0	114	10000	57828516545.0500	71563566909.3600	23.7513 *
10_District1	571	10000	8018849193135.5100	8218857439409.0600	2.4942 *
10_District2	578	10000	40625364513.9699	27771011394.0200	46.2869
10_District3	247	10000	129045115417.5300	139155792160.4600	7.8349 *
10_District4	1935	10000	24789226229665.7030	21291228411410.1000	16.4292
10_District5	1541	10000	39368704514405.7000	37249297856014.8000	5.6897
11_District0	143	10000	3659838650.6500	3046672128.3800	20.1257
11_District1	1160	10000	6534453176402.4795	6157994611789.7000	6.1133
11_District2	199	10000	12529143717.1300	9214556192.4000	35.9712
11_District3	200	10000	58426923692.1299	61007292651.7800	4.4164 *
11_District4	2190	10000	24653411632109.7270	22460659385239.4000	9.7626
11_District5	1469	10000	15976061388576.6970	15643005576898.3000	2.1291
12_District0	164	10000	127332780450.3600	115036288619.9000	10.6892
12_District1	1717	10000	41874084461133.0900	37877356021322.0000	10.5517
12_District2	241	10000	149496005456.4600	153054550706.1000	2.3803 *
12_District3	595	10000	212347162085.9698	239836499336.7000	12.9454 *
12_District4	2942	10000	64672333500298.7800	59129219661289.8000	9.3745
12_District5	2901	10000	72328352734637.8400	65945373929847.0000	9.6791
13_District0	80	10000	176478198913.1201	154315121626.0000	14.3622
13_District1	853	10000	17465242494990.0400	14674175609787.1000	19.0202
13_District2	316	10000	116084153493.2699	126837068235.0100	9.2630 *
13_District3	1069	10000	418869015611.6208	429665639491.2800	2.5775 *
13_District4	1358	10000	30680129989003.6700	30033656135618.7000	2.1524
13_District5	805	10000	46949266940678.5100	42764648834839.3000	9.7852
14_District0	74	10000	33613834518.4900	34977146773.4600	4.0558 *
14_District1	1877	10000	13122494080941.2320	12434388027267.4000	5.5338
14_District2	187	10000	93003835894.4398	90165619840.7900	3.1477
14_District3	701	10000	222880056492.5500	262406322982.9800	17.7343 *
14_District4	1738	10000	38513662440545.7340	31546738861338.4000	22.0844
14_District5	1381	10000	46381959868001.5900	44524141233501.8000	4.1726
15_District0	63	10000	54827547254.4899	42188641727.2300	29.9580
15_District1	980	10000	12602573286764.2230	12317798430422.9000	2.3118
15_District2	147	10000	76110681486.7102	67421894826.9300	12.8871
15_District3	516	10000	386925996569.5306	463823619193.1600	19.8739 *
15_District4	1905	10000	27434068605268.5550	22834329911998.4000	20.1439
15_District5	1366	10000	31234193557599.5270	28700034523627.8000	8.8298
16_District0	76	10000	123379622136.1200	120807470457.0500	2.1291
16_District1	828	10000	13413700712982.5060	12316160134202.9000	8.9113
16_District2	156	10000	97539952435.7500	94504646913.7700	3.2118
16_District3	577	10000	214464159387.2001	210518442436.7200	1.8742
16_District4	1385	10000	31192659518846.5270	28327769996973.5000	10.1133
16_District5	1475	10000	49451132314209.8400	48522145053303.8000	1.9145
17_District0	52	10000	64524717068.3099	60633779564.4100	6.4171
17_District1	1479	10000	11617082853705.7950	12050832937058.4000	3.7337 *
17_District2	290	10000	115751341239.7699	111787046906.6000	3.5462
17_District3	159	10000	181179309939.1000	178730940805.7800	1.3698
17_District4	1609	10000	20618254258146.6250	19092974442252.1000	7.9886
17_District5	820	10000	26652126436980.3750	23886759047749.6000	11.5769
18_District0	24	10000	2710365389.5300	2341527488.6000	15.7520
18_District1	354	10000	13646938890431.1250	13924688690304.7000	2.0352 *
18_District2	166	10000	5766218027.7500	4417358464.4100	30.5354
18_District3	100	10000	55168307609.2599	70920528499.0700	28.5530 *

Table E.1 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
18_District4	1447	10000	18285427769696.9770	18718722507656.5000	2.3696 *
18_District5	512	10000	14346956986764.6860	12641617614178.6000	13.4898
19_District0	72	10000	47758552496.3500	55462969817.4300	16.1320 *
19_District1	1186	10000	24618865063621.6100	22401449014876.9000	9.8985
19_District2	361	10000	159323653542.6700	153280470262.9000	3.9425
19_District3	316	10000	219558314420.6599	243569037679.2000	10.9359 *
19_District4	2372	10000	120228763973581.9700	106192108824987.0000	13.2181
19_District5	1645	10000	86255205279477.3600	78807991381031.8000	9.4498
1_District0	86	10000	425000131834.3106	430590936121.1900	1.3154 *
1_District1	1912	10000	56462353644529.3100	56270206789859.1000	0.3414
1_District2	208	10000	623580041674.7792	672001346398.5800	7.7650 *
1_District3	805	10000	1012009511413.5295	1031045184732.7300	1.8809 *
1_District4	3607	10000	82173119516438.2000	73732177837593.4000	11.4481
1_District5	1941	10000	94192565467190.9700	91806570717137.9000	2.5989
20_District0	44	10000	98281656368.8800	98421186256.3800	0.1419 *
20_District1	877	10000	17808835626053.7700	17222315357479.6000	3.4055
20_District2	443	10000	29944665225.3700	27378465012.0200	9.3730
20_District3	981	10000	262346802562.0587	305213925793.8100	16.3398 *
20_District4	2047	10000	28506210512191.0470	24960379220572.1000	14.2058
20_District5	543	3464	48088160922901.7200	48965990159239.8000	1.8254 *
21_District0	366	10000	108332912943.4300	120414278382.3300	11.1520 *
21_District1	572	10000	17721038066214.7400	16893868904739.4000	4.8962
21_District2	642	10000	80707473855.3998	62666013838.2000	28.7898
21_District3	477	10000	843250669428.9386	892017226174.9800	5.7831 *
21_District4	1865	10000	25298959861932.4600	24304758516183.0000	4.0905
21_District5	1071	10000	49699318164858.3500	45930297567746.2000	8.2059
22_District0	53	10000	146651795790.9399	138906549645.3000	5.5758
22_District1	440	10000	15517247869615.6910	14071948171020.4000	10.2707
22_District2	159	10000	232904094204.2603	226837888943.8600	2.6742
22_District3	720	10000	279269051966.6899	285892462254.7700	2.3716 *
22_District4	1327	10000	32239196238177.2400	29074924558505.3000	10.8831
22_District5	1365	10000	44999254901547.8050	42772519577575.2000	5.2059
23_District0	64	10000	35453041497.5900	32906581132.3800	7.7384
23_District1	1581	10000	18553983673921.5980	17211576588480.4000	7.7994
23_District2	130	10000	197577784164.8399	183871311138.5000	7.4543
23_District3	677	10000	213132901005.6505	250619333021.8100	17.5882 *
23_District4	1762	10000	38395710924473.5300	34555727151866.1000	11.1124
23_District5	1515	10000	76698336949283.1600	70608343796642.5000	8.6250
24_District0	116	10000	124292349244.8500	105830236823.5000	17.4450
24_District1	832	10000	13134075969512.0160	11389191027676.9000	15.3205
24_District2	401	10000	23856949683.1099	26055867878.7800	9.2170 *
24_District3	1035	10000	101838532588.1899	159315579521.2200	56.4393 *
24_District4	2196	10000	25972810944861.7500	24517137738123.8000	5.9373
24_District5	744	10000	26911410512129.4200	24738655441667.6000	8.7828
25_District0	29	10000	1616800882.2799	1255504790.1000	28.7769
25_District1	1265	10000	6031913668330.9880	6071232244353.8300	0.6518 *
25_District2	69	10000	2789641143.3100	2430507030.7800	14.7760
25_District3	191	10000	53545669591.4900	54825213021.3900	2.3896 *
25_District4	1706	10000	22993087927410.2400	20503174454937.0000	12.1440
25_District5	706	10000	11093840620670.7250	11011719621667.4000	0.7457
26_District0	177	10000	215785899433.4600	225380467772.9800	4.4463 *
26_District1	970	10000	44444715105126.6250	42180173025536.8000	5.3687
26_District2	183	10000	341154670791.7099	282951685653.1000	20.5699
26_District3	1662	10000	1010951700327.3400	1171533328289.0400	15.8842 *
26_District4	3004	10000	114416918081598.1200	105580668249051.0000	8.3691
26_District5	2032	10000	78563574443481.4400	76363123397244.6000	2.8815
27_District0	144	10000	141123569491.1200	162479691641.3000	15.1329 *
27_District1	1058	10000	20160867712462.1050	18425601260045.1000	9.4176
27_District2	267	10000	64505386329.4599	61192189969.0700	5.4144
27_District3	507	10000	129500504280.6899	116712931257.8400	10.9564
27_District4	1621	10000	37680793520222.9800	33778988955119.3000	11.5509
27_District5	1488	10000	39813786783293.8200	39568513214098.5000	0.6198
28_District0	46	10000	124239127798.0099	102108349093.8000	21.6738
28_District1	1182	10000	14761964185111.9840	15128335139531.3000	2.4818 *
28_District2	196	10000	68044935657.5799	64692968024.3600	5.1813
28_District3	648	10000	740018619541.0001	805624097687.0900	8.8653 *
28_District4	1997	10000	25882040182480.9960	22982825117023.6000	12.6147
28_District5	1103	10000	36173595121688.8100	33791534896023.1000	7.0492
29_District0	231	10000	33389137635.3299	29353316624.4500	13.7491
29_District1	344	10000	7664589729242.2970	7484826255380.2500	2.4017
29_District2	364	10000	78961985344.2499	101233312043.5700	28.2051 *
29_District3	771	10000	225937596356.0400	232601979076.2900	2.9496 *
29_District4	1301	10000	9550657561663.9160	8309110124637.3200	14.9420
29_District5	2566	10000	36360343783848.0100	35670467010142.6000	1.9340
2_District0	100	10000	143938419743.6200	147005506881.3000	2.1308 *
2_District1	1131	10000	42646555520397.6100	39540340925649.8000	7.8558
2_District2	492	10000	194240608635.1301	217829855689.6200	12.1443 *
2_District3	347	10000	840013613778.2102	858787147589.8100	2.2349 *
2_District4	4076	10000	65211203946622.9840	58876326066126.1000	10.7596
2_District5	1745	10000	103739457393397.2500	95903342380824.9000	8.1708
30_District0	48	10000	15519711254.6900	14856579555.0000	4.4635
30_District1	606	10000	4186035210142.2983	4288991390631.0100	2.4595 *
30_District2	354	10000	37878747544.7700	28984654012.0000	30.6855
30_District3	682	10000	119336540892.1000	140260859804.8700	17.5338 *
30_District4	1413	10000	5842803029893.0070	6033193692111.0700	3.2585 *
30_District5	417	10000	5439541091348.2080	5491783393822.4800	0.9604 *
3_District0	63	10000	97411177384.8300	90701681116.3600	7.3973
3_District1	740	10000	18282156558812.9920	17931112587450.7000	1.9577
3_District2	247	10000	173531521228.0999	184423936337.4900	6.2769 *
3_District3	879	10000	821302276118.0098	837894240881.4400	2.0202 *
3_District4	4411	10000	77785891987793.1200	67042838268903.4000	16.0241
3_District5	2458	10000	64469278292389.0100	60566955554266.6000	6.4429
4_District0	32	10000	21991284743.1700	20379141775.5100	7.9107
4_District1	1315	10000	16663241257517.7420	16091560843817.3000	3.5526
4_District2	190	10000	20912680421.4499	19771989271.8700	5.7692
4_District3	226	10000	29070907160.0599	31243041764.1500	7.4718 *
4_District4	3331	10000	49362523927800.7900	44497725128783.0000	10.9326

Table E.1 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
4_District5	1422	10000	32047671081430.7100	30292083288905.3000	5.7955
5_District0	229	10000	108361452677.7999	105305055950.1000	2.9024
5_District1	883	10000	23293453038218.5400	22578812619408.6000	3.1650
5_District2	179	10000	72814721759.8400	49746631100.9200	46.3711
5_District3	528	10000	297355502880.3797	366270241695.4900	23.1758 *
5_District4	2473	10000	79778112199245.7500	68243049339872.1000	16.9029
5_District5	1500	10000	128453320339184.8400	121803986176569.0000	5.4590
6_District0	92	10000	174240677714.6999	174024536017.0000	0.1242
6_District1	1184	10000	26209123692126.2930	24635464198083.6000	6.3877
6_District2	332	10000	238556186590.7401	263920214770.7000	10.6323 *
6_District3	396	10000	266599152914.4895	272026860685.1900	2.0359 *
6_District4	1547	10000	35768389360388.9900	32671893526336.1000	9.4775
6_District5	1438	10000	56765748560626.7900	52577668995018.6000	7.9655
7_District0	274	10000	48036330723.9400	53992400830.1800	12.3990 *
7_District1	1709	10000	25689484175715.2540	25530776489059.8000	0.6216
7_District2	175	10000	108633497561.8599	96812988604.0900	12.2096
7_District3	361	10000	650264476114.4296	627921570619.9200	3.5582
7_District4	1605	10000	33698871042998.8050	30888008083772.9000	9.1001
7_District5	1897	10000	53210556107282.3700	52214576657422.1000	1.9074
8_District0	55	10000	57350067798.4000	47815962594.9100	19.9391
8_District1	1436	10000	14344465830917.8200	13654929607346.6000	5.0497
8_District2	336	10000	69729215441.7400	73150043166.7900	4.9058 *
8_District3	744	10000	356144436938.8400	367102727288.8900	3.0769 *
8_District4	2156	10000	32072106246520.2030	29816748603227.9000	7.5640
8_District5	761	10000	42646298319684.1900	40450402089088.4000	5.4286
9_District0	48	10000	102701631556.8100	100093204504.3000	2.6059
9_District1	625	10000	11923400674846.3460	12178217016824.2000	2.1371 *
9_District2	120	10000	97715845042.7999	89342594098.5600	9.3720
9_District3	1044	10000	983125684210.9888	911115793606.4100	7.9034
9_District4	2328	10000	42073285949982.7700	37164202673658.1000	13.2091
9_District5	751	10000	28092914049076.2800	25382936280280.4000	10.6763
C101_100t_20w	771	10000	4085430108.3999	-12304901708.1200	133.2016
C101_25t_5w	30	10000	111640260.9600	9512566.6400	1073.6081
C101_50t_10w	128	10000	-362313973.1799	-1079154704.1200	66.4261
C102_100t_20w	1338	10000	33315667041.5800	-5155408825.8200	746.2274
C102_25t_5w	74	10000	45557675.5002	-37546115.7600	221.3379
C102_50t_10w	264	10000	136358219.3198	-1102530180.6200	112.3677
C103_100t_20w	3111	10000	10602654869.2802	-7222438475.8600	246.8015
C103_25t_5w	218	10000	36546542.7998	-43052838.4600	184.8876
C103_50t_10w	1079	10000	167525534.1999	-1110321792.2400	115.0880
C104_100t_20w	4272	10000	2869531113.5005	-8681519205.2400	133.0533
C104_25t_5w	349	10000	35545370.4999	-33540706.8600	205.9768
C104_50t_10w	2413	10000	-416855888.2400	-1086946666.3200	61.6489
C105_100t_20w	1248	10000	-10578323043.2999	-12158993843.3000	13.0000
C105_25t_5w	36	10000	155695012.0000	1002190.7200	15435.4673
C105_50t_10w	270	10000	-377897357.8800	-1102529614.6600	65.7245
C106_100t_20w	2183	10000	-10943092987.0601	-5349954009.7600	51.1111 *
C106_25t_5w	48	10000	114143319.8000	9512566.6400	1099.9213
C106_50t_10w	216	10000	-868777801.6400	-1086946589.7000	20.0717
C107_100t_20w	1947	10000	-10578323327.5801	-6979258964.6200	34.0230 *
C107_25t_5w	55	10000	112140921.2800	-2001420.3200	5703.0669
C107_50t_10w	481	10000	-981757674.7002	-1125905161.9000	12.8028
C108_100t_20w	2269	10000	2942485491.1397	-6492899831.1000	145.3185
C108_25t_5w	59	10000	113142097.8800	-8009072.9800	1512.6740
C108_50t_10w	586	10000	-958382260.4999	-1125905177.3200	14.8789
C109_100t_20w	3699	10000	10991743079.2599	-6614488230.8200	266.1767
C109_25t_5w	62	10000	38048331.6798	-26031566.9600	246.1622
C109_50t_10w	821	10000	-935007224.7601	-1125905761.1800	16.9551
C201_100t_20w	7200	8011	-12596716053.9600	-12596720162.2300	0.0000
C201_25t_5w	384	10000	-42552163.2401	-45555916.9200	6.5935
C201_50t_10w	2141	10000	-1110321494.1802	-1125905241.3800	1.3841
C202_100t_20w	7200	5602	17849419939.7999	-6055175633.1600	394.7795
C202_25t_5w	2700	10000	-42552189.2800	-45555950.8800	6.5935
C202_50t_10w	7200	8866	89607610.5598	-1125905361.9600	107.9587
C203_100t_20w	7202	4286	17411695819.7996	-6930623788.8800	351.2284
C203_25t_5w	7200	7033	-42552175.3800	-45555956.2400	6.5936
C203_50t_10w	7207	5601	155838211.7600	-1125905456.0600	113.8411
C204_100t_20w	7203	2565	2820895347.0999	-8657198950.2400	132.5843
C204_25t_5w	7204	4784	-45555856.0201	-45555977.0000	0.0002
C204_50t_10w	7201	2124	93504394.2998	-1125905585.1400	108.3048
C205_100t_20w	7201	5347	9702889586.1598	-12596718687.8400	177.0271
C205_25t_5w	974	10000	-42552170.0398	-45555926.8000	6.5935
C205_50t_10w	6220	10000	-1125904887.6001	-1125905403.9800	0.0000
C206_100t_20w	7202	2659	17460331452.5199	-7733117210.9600	325.7864
C206_25t_5w	1357	10000	-44053957.9600	-45555926.8000	3.2969
C206_50t_10w	7200	8220	-479190002.2399	-1125905541.4200	57.4395
C207_100t_20w	7201	2165	24488233802.3396	-7222438653.2400	439.0576
C207_25t_5w	1477	10000	-45555894.9198	-45555933.4200	0.0000
C207_50t_10w	7201	4252	-1067466345.9200	-1125905455.9000	5.1904
C208_100t_20w	7200	3075	17314423481.6797	-8657201177.3800	300.0002
C208_25t_5w	1945	10000	-42552170.0396	-45555933.1600	6.5935
C208_50t_10w	7200	7326	-1102529619.5000	-1125905465.6600	2.0761
hh_00_P0	865	10000	50080851377.5085	6663651008.2900	651.5527
ll1_00_P0	162	10000	373710720.3599	1338732826.5000	258.2270 *
ll1_01_P0	160	10000	3626241564.0199	1338732826.5000	170.8711
ll1_02_P0	157	10000	373768965.3901	1338732826.5000	258.1712 *
ll1_03_P0	160	10000	373695618.3599	1338732826.5000	258.2415 *
ll1_04_P0	174	10000	342558398.3900	1338732826.5000	290.8042 *
ll1_05_P0	160	10000	373716559.8599	1307660206.1200	249.9069 *
ll1_06_P0	154	10000	5369184942.1897	1214281948.4400	342.1695
ll1_07_P0	167	10000	342574210.2399	1338732826.5000	290.7862 *
ll2_00_P0	229	10000	-179579601.5000	85352916.2700	147.5292 *
ll3_00_P0	199	10000	-179674136.5400	-198615845.6200	9.5368
R101_100t_20w	59	10000	21008053672.5398	6319981502.8600	232.4068
R101_25t_5w	5	10000	60631893.5799	47893886.5800	26.5963
R101_50t_10w	18	10000	1170928743.0599	709051142.3992	65.1402
R102_100t_20w	76	10000	16071499804.3999	14101742003.1800	13.9681

Table E.1 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
R102_25t_5w	8	10000	38715848.3199	34655177.0000	11.7173
R102_50t_10w	17	10000	1091712944.1399	641955625.9388	70.0604
R103_100t_20w	108	10000	12764252097.5999	12469734030.7400	2.3618
R103_25t_5w	10	10000	34321551.5800	30260803.3400	13.4191
R103_50t_10w	28	10000	828525009.6399	570531480.7600	45.2198
R104_100t_20w	178	10000	11143052551.4199	14023383931.9600	25.8486 *
R104_25t_5w	9	10000	25644010.7200	29982663.2800	16.9187 *
R104_50t_10w	51	10000	500405798.4199	293491458.8800	70.5009
R105_100t_20w	107	10000	14444895627.9399	8424839701.0800	71.4560
R105_25t_5w	8	10000	43332593.8799	35155829.6400	23.2586
R105_50t_10w	21	10000	902113560.6799	388290958.0600	132.3292
R106_100t_20w	101	10000	13588361915.1399	8443754034.7200	60.9279
R106_25t_5w	7	10000	34710842.4200	30483330.4600	13.8682
R106_50t_10w	27	10000	834152346.9199	303880567.4400	174.5000
R107_100t_20w	113	10000	10362174097.5600	11634816307.3400	12.2816 *
R107_25t_5w	11	10000	30038339.4199	22028382.6800	36.3619
R107_50t_10w	33	10000	639358534.8199	372274723.4800	71.7437
R108_100t_20w	153	10000	7846612205.3599	12369760088.0800	57.6445 *
R108_25t_5w	11	10000	26088948.8999	26033406.5200	0.2133
R108_50t_10w	52	10000	501271656.7199	293491505.9600	70.7959
R109_100t_20w	157	10000	10372982089.3999	6636116061.1200	56.3110
R109_25t_5w	7	10000	42609637.8599	26255954.0600	62.2856
R109_50t_10w	31	10000	837615379.6799	631133985.4200	32.7159
R110_100t_20w	113	10000	12785868019.5399	7622345982.8200	67.7419
R110_25t_5w	8	10000	38771534.5599	34488424.6600	12.4189
R110_50t_10w	23	10000	647583256.5199	636761297.5800	1.6995
R111_100t_20w	110	10000	11094416005.7598	8473476077.2600	30.9311
R111_25t_5w	9	10000	34432735.2998	26144634.7600	31.7009
R111_50t_10w	23	10000	508630350.5000	496942686.1600	2.3519
R112_100t_20w	126	10000	7962798075.4399	10851236130.0600	36.2741 *
R112_25t_5w	9	10000	34543997.2000	26311467.9000	31.2887
R112_50t_10w	23	10000	500838587.3200	427249792.1800	17.2238
R201_100t_20w	2547	10000	-559307234.3001	-1383419122.5000	59.5706
R201_25t_5w	126	10000	-5171448.4601	-5171519.4600	0.0013
R201_50t_10w	1362	10000	-124231486.3201	-125097475.6400	0.6922
R202_100t_20w	1816	10000	3607176575.4799	-778169509.0800	563.5463
R202_25t_5w	219	10000	-721637.5801	-5171550.1800	86.0460
R202_50t_10w	909	10000	-48045138.3400	-125097689.9000	61.5939
R203_100t_20w	2640	10000	281014374.1999	-891653653.5400	131.5160
R203_25t_5w	470	10000	-721702.1600	-5171614.5200	86.0449
R203_50t_10w	2016	10000	-57135809.4399	-125097805.5400	54.3270
R204_100t_20w	3624	10000	310736292.3799	-1007840112.0800	130.8319
R204_25t_5w	614	10000	-4893340.9200	-5060450.1000	3.3022
R204_50t_10w	5972	10000	-55837307.8400	-125098025.1800	55.3651
R205_100t_20w	4626	10000	1896810807.6597	-910567724.8000	308.3107
R205_25t_5w	259	10000	-5171520.0602	-5171759.2000	0.0046
R205_50t_10w	2305	10000	-58001601.0200	-125097876.3800	53.6350
R206_100t_20w	3791	10000	1110528318.1999	-964607487.1800	215.1274
R206_25t_5w	423	10000	-5171712.6401	-5171813.9000	0.0019
R206_50t_10w	1621	10000	-54971339.7000	-125097837.4400	56.0573
R207_100t_20w	3045	10000	316140108.1799	-1018647805.1000	131.0352
R207_25t_5w	593	10000	-5171742.7599	-5171794.9600	0.0010
R207_50t_10w	2026	10000	-57136009.2400	-125097967.7600	54.3269
R208_100t_20w	4138	10000	-489055634.4402	-1042965700.1000	53.1091
R208_25t_5w	702	10000	-5060287.4001	-5060560.5800	0.0053
R208_50t_10w	5020	10000	-58867300.2800	-125098091.9200	52.9430
R209_100t_20w	6006	10000	1937340343.6198	-1013243421.1000	291.2018
R209_25t_5w	342	10000	-610329.6604	-5060483.0600	87.9393
R209_50t_10w	2645	10000	-112976783.5400	-125097999.8800	9.6893
R210_100t_20w	3799	10000	3572050759.5198	-978117773.2200	465.1963
R210_25t_5w	394	10000	-4615070.5800	-5171642.3200	10.7619
R210_50t_10w	2262	10000	-58001444.7599	-125097934.6800	53.6351
R211_100t_20w	4431	10000	5214866174.4999	-942991911.9800	653.0128
R211_25t_5w	456	10000	-610510.4602	-5060502.8400	87.9357
R211_50t_10w	2594	10000	-115573874.4200	-122500836.0400	5.6546
RC101_100t_20w	57	10000	14558380234.0199	15814809991.8800	8.6302 *
RC101_25t_5w	9	10000	39438835.2199	27034405.3800	45.8838
RC101_50t_10w	19	10000	649314738.1999	190034189.6200	241.6831
RC102_100t_20w	66	10000	16047182306.6399	15822916717.4800	1.4173
RC102_25t_5w	13	10000	34655172.0400	26645148.4800	30.0618
RC102_50t_10w	27	10000	773549852.1600	578756222.9600	33.6572
RC103_100t_20w	96	10000	11999586723.9598	17527878844.9600	46.0706 *
RC103_25t_5w	12	10000	30427814.0799	21805915.8800	39.5392
RC103_50t_10w	22	10000	647150488.0000	767922882.8400	18.6621 *
RC104_100t_20w	124	10000	8770696590.4398	18130424572.1200	106.7159 *
RC104_25t_5w	12	10000	25977730.7799	25810879.2400	0.6464
RC104_50t_10w	38	10000	700827236.8199	498674346.2200	40.5380
RC105_100t_20w	59	10000	16117434518.3998	10907978098.4200	47.7582
RC105_25t_5w	10	10000	35378238.0600	30761388.4200	15.0085
RC105_50t_10w	22	10000	848870062.2400	448893536.2400	89.1027
RC106_100t_20w	88	10000	11294364348.9799	11529438265.0600	2.0813 *
RC106_25t_5w	11	10000	35044583.0599	22417682.5200	56.3256
RC106_50t_10w	21	10000	648016216.9199	515989390.5800	25.5871
RC107_100t_20w	96	10000	13693740456.3399	16638920438.6600	21.5074 *
RC107_25t_5w	8	10000	30872691.8199	30761589.8800	0.3611
RC107_50t_10w	19	10000	454521045.1000	575726019.2200	26.6665 *
RC108_100t_20w	87	10000	11267344669.8199	14823176358.6800	31.5587 *
RC108_25t_5w	10	10000	26533963.5399	30372205.5600	14.4653 *
RC108_50t_10w	37	10000	448460712.6400	630268451.2000	40.5403 *
RC201_100t_20w	2254	10000	-1388819673.9802	-1378013035.7800	0.7781 *
RC201_25t_5w	68	10000	-5059580.0600	-5171005.4000	2.1548
RC201_50t_10w	771	10000	-121632755.5400	-125096456.0400	2.7688
RC202_100t_20w	2036	10000	1923832352.4199	-848419159.5200	326.7549
RC202_25t_5w	99	10000	-5170964.3200	-5171480.4200	0.0099
RC202_50t_10w	808	10000	77921659.6798	-125096955.0600	162.2890
RC203_100t_20w	2647	10000	291823679.6598	-915968950.1800	131.8595
RC203_25t_5w	366	10000	-721545.6600	-5171505.3800	86.0476

Table E.1 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
RC203_50t_10w	1111	10000	10826455.9398	-125096765.1800	108.6544
RC204_100t_20w	3943	10000	1105126050.0798	-1032155821.9000	207.0696
RC204_25t_5w	582	10000	-5060302.6799	-5060413.3600	0.0021
RC204_50t_10w	3151	10000	-38954116.8399	-125097014.9000	68.8608
RC205_100t_20w	2737	10000	289121564.3799	-856525543.8000	133.7551
RC205_25t_5w	81	10000	-4893158.1800	-5171462.0600	5.3815
RC205_50t_10w	1011	10000	77922083.9997	-125096766.4200	162.2894
RC206_100t_20w	3427	10000	1078106455.3398	-905161565.1000	219.1065
RC206_25t_5w	166	10000	-5171260.1601	-5171397.9400	0.0026
RC206_50t_10w	1515	10000	-121200050.8400	-125097137.7600	3.1152
RC207_100t_20w	4566	10000	1931937210.5599	-907864881.6600	312.8000
RC207_25t_5w	338	10000	-4892923.3203	-5059890.2000	3.2998
RC207_50t_10w	1499	10000	-119468348.1199	-122499526.2000	2.4744
RC208_100t_20w	3486	10000	4409671916.6798	-978116228.2000	550.8331
RC208_25t_5w	339	10000	-5060424.6200	-5060744.0000	0.0063
RC208_50t_10w	2131	10000	-52806087.2000	-125096819.1800	57.7878
test150-0-0-0-0_d0_tw0	639	719	69666303241.3001	-28349336446.9000	345.7422
test150-0-0-0-0_d0_tw1	4944	10000	-55870971638.1994	-30832491493.7000	44.8148 *
test150-0-0-0-0_d0_tw2	3292	10000	-24348699719.7993	-28832172176.8000	15.5502
test150-0-0-0-0_d0_tw3	2900	10000	-24348699747.5998	-27383664735.0000	11.0831
test150-0-0-0-0_d0_tw4	2326	10000	69942208924.7995	-26142085667.0000	367.5463
test250-0-0-0-0_d0_tw0	-	-	-	23650649218.5000	-
test250-0-0-0-0_d0_tw1	2030	3155	-214544938387.3036	-292254131472.0000	26.5895
test250-0-0-0-0_d0_tw2	1890	2977	40543948303.6960	-275698693917.1000	114.7058
test250-0-0-0-0_d0_tw3	6083	10000	-469633821865.2001	-266914175565.0000	43.1654 *
test250-0-0-0-0_d0_tw4	6359	10000	1315988380894.5950	-231776103927.7000	667.7843
test50-0-0-0-0_d0_tw0	1043	5090	-842598576.7999	-842599324.2000	0.0000
test50-0-0-0-0_d0_tw1	996	10000	-842598443.2999	-842599506.6000	0.0001
test50-0-0-0-0_d0_tw2	949	10000	-842598048.7999	-842599560.3000	0.0001
test50-0-0-0-0_d0_tw3	964	10000	-836355937.1999	-842598590.0000	0.7408
test50-0-0-0-0_d0_tw4	961	10000	-823873632.4999	-842599753.8000	2.2224

E.2 Tabu Search Results: Config. 2

Table E.2: Tabu Search experiments results with parameter configuration 2

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
10_District0	19	1000	71023308535.1199	71563566909.3600	0.7606 *
10_District1	166	1000	8674802167667.0470	8218857439409.0600	5.5475
10_District2	139	1000	46572004135.2700	27771011394.0200	67.7000
10_District3	35	1000	142335475606.1500	139155792160.4600	2.2849
10_District4	871	1000	26624756170246.7900	21291228411410.1000	25.0503
10_District5	389	1000	40802332448450.5500	37249297856014.8000	9.5385
11_District0	13	1000	3631096474.7400	3046672128.3800	19.1823
11_District1	184	1000	6862357325571.0090	6157994611789.7000	11.4381
11_District2	41	1000	13646633113.0399	9214556192.4000	48.0986
11_District3	45	1000	65476859781.5200	61007292651.7800	7.3262
11_District4	557	1000	25297528070007.2070	22460659385239.4000	12.6303
11_District5	301	1000	16444910089646.0900	15643005576898.3000	5.1262
12_District0	17	1000	127522687343.8999	115036288619.9000	10.8543
12_District1	328	1000	43715751249704.6400	37877356021322.0000	15.4139
12_District2	57	1000	169105076309.3399	153054550706.1000	10.4868
12_District3	101	1000	242207056403.2100	239836499336.7000	0.9884
12_District4	994	1000	66178019038127.0700	59129219661289.8000	11.9210
12_District5	697	1000	74587481844678.5600	65945373929847.0000	13.1049
13_District0	13	1000	180760555789.0700	154315121626.0000	17.1372
13_District1	217	1000	18673126631813.0700	14674175609787.1000	27.2516
13_District2	70	1000	128595407330.7600	126837068235.0100	1.3862
13_District3	258	1000	434530784343.2900	429665639491.2800	1.1323
13_District4	383	1000	33652849914880.6640	30033656135618.7000	12.0504
13_District5	340	1000	50269122425108.5860	42764648834839.3000	17.5483
14_District0	16	1000	33657460214.3500	34977146773.4600	3.9209 *
14_District1	466	1000	13738870890562.8180	12434388027267.4000	10.4909
14_District2	36	1000	108150942438.9300	90165619840.7900	19.9469
14_District3	101	1000	264510383345.8802	262406322982.9800	0.8018
14_District4	621	1000	39955847900997.1640	31546738861338.4000	26.6560
14_District5	558	1000	48571301176060.1640	44524141233501.8000	9.0898
15_District0	14	1000	52541932428.5599	42188641727.2300	24.5404
15_District1	201	1000	13343268477116.9280	12317798430422.9000	8.3251
15_District2	52	1000	85366688281.5999	67421894826.9300	26.6156
15_District3	123	1000	452400421567.4308	463823619193.1600	2.5250 *
15_District4	559	1000	28593564029826.0940	22834329911998.4000	25.2218
15_District5	481	1000	31608642685943.7850	28700034523627.8000	10.1345
16_District0	8	1000	123672387191.8699	120807470457.0500	2.3714
16_District1	156	1000	14804226644434.3100	12316160134202.9000	20.2016
16_District2	25	1000	107138080055.4999	94504646913.7700	13.3680
16_District3	84	1000	263593147352.4600	210518442436.7200	25.2114
16_District4	608	1000	35518712315877.2400	28327769996973.5000	25.3847
16_District5	514	1000	52936772235343.2500	48522145053303.8000	9.0981
17_District0	7	1000	66735476864.7799	60633779564.4100	10.0631
17_District1	321	1000	12330487596968.9340	12050832937058.4000	2.3206
17_District2	49	1000	110909455460.6700	111787046906.6000	0.7912 *
17_District3	26	1000	214450895802.3900	178730940805.7800	19.9853
17_District4	481	1000	21612019261830.1520	19092974442252.1000	13.1935
17_District5	211	1000	27944804202110.6170	23886759047749.6000	16.9886
18_District0	3	1000	2869798540.3400	2341527488.6000	22.5609
18_District1	95	1000	14590792231270.1580	13924688690304.7000	4.7836

Table E.2 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
18_District2	27	1000	6936082119.9099	4417358464.4100	57.0187
18_District3	15	1000	62108760133.0200	70920528499.0700	14.1876 *
18_District4	422	1000	20299891016845.2230	18718722507656.5000	8.4469
18_District5	111	1000	14949615499196.6680	12641617614178.6000	18.2571
19_District0	10	1000	50100813233.8200	55462969817.4300	10.7027 *
19_District1	234	1000	25523960449110.1880	22401449014876.9000	13.9388
19_District2	53	1000	159808599681.2900	153280470262.9000	4.2589
19_District3	75	1000	237044866898.2397	243569037679.2000	2.7522 *
19_District4	1366	1000	129861724782360.6700	106192108824987.0000	22.2894
19_District5	465	1000	90638153482897.6400	78807991381031.8000	15.0113
1_District0	20	1000	435538551689.1299	430590936121.1900	1.1490
1_District1	594	1000	57742532072808.1640	56270206789859.1000	2.6165
1_District2	63	1000	681386608176.6498	672001346398.5800	1.3966
1_District3	214	1000	1040724340494.9508	1031045184732.7300	0.9387
1_District4	1596	1000	88015986409409.8800	73732177837593.4000	19.3725
1_District5	850	1000	97125915546721.8300	91806570717137.9000	5.7940
20_District0	7	1000	98316539195.2500	98421186256.3800	0.1064 *
20_District1	235	1000	18751313021996.2770	17222315357479.6000	8.8780
20_District2	68	1000	34822143743.5299	27378465012.0200	27.1880
20_District3	244	1000	286404352249.9301	305213925793.8100	6.5674 *
20_District4	805	1000	29727988816359.0500	24960379220572.1000	19.1007
20_District5	440	1000	49889961704805.5860	48965990159239.8000	1.8869
21_District0	59	1000	113175444271.1200	120414278382.3300	6.3961 *
21_District1	233	1000	18434629214254.9060	16893868904739.4000	9.1202
21_District2	71	1000	103895813624.2799	62666013838.2000	65.7929
21_District3	89	1000	949631279962.5399	892017226174.9800	6.4588
21_District4	503	1000	26831351238992.0700	24304758516183.0000	10.3954
21_District5	318	1000	51393689812545.8500	45930297567746.2000	11.8949
22_District0	8	1000	155844514136.7699	138906549645.3000	12.1937
22_District1	129	1000	16185901813535.5180	14071948171020.4000	15.0224
22_District2	30	1000	261372272563.0800	226837888943.8600	15.2242
22_District3	131	1000	304025405199.1001	285892462254.7700	6.3425
22_District4	434	1000	33915717191232.0230	29074924558505.3000	16.6493
22_District5	368	1000	46033649335446.8050	42772519577575.2000	7.6243
23_District0	8	1000	35383951488.5700	32906581132.3800	7.5284
23_District1	322	1000	19300357279581.7400	17211576588480.4000	12.1359
23_District2	23	1000	215007507921.3901	183871311138.5000	16.9336
23_District3	128	1000	234574941983.0599	250619333021.8100	6.8397 *
23_District4	612	1000	40999719262551.8600	34555727151866.1000	18.6481
23_District5	525	1000	78307969430628.6600	70608343796642.5000	10.9046
24_District0	21	1000	124447493183.3399	105830236823.5000	17.5916
24_District1	169	1000	13876242045338.0060	11389191027676.9000	21.8369
24_District2	66	1000	27446750713.6800	26055867878.7800	5.3380
24_District3	152	1000	153837901790.0100	159315579521.2200	3.5606 *
24_District4	627	1000	28069693758170.2000	24517137738123.8000	14.4900
24_District5	174	1000	27586726276307.8320	24738655441667.6000	11.5126
25_District0	5	1000	1788416175.1100	1255504790.1000	42.4459
25_District1	179	1000	6680036021220.9000	6071232244353.8300	10.0276
25_District2	11	1000	3039947227.6099	2430507030.7800	25.0746
25_District3	38	1000	53437233714.0100	54825213021.3900	2.5974 *
25_District4	445	1000	23986888174820.2420	20503174454937.0000	16.9910
25_District5	167	1000	11758314299630.1000	11011719621667.4000	6.7800
26_District0	36	1000	237364488936.3699	225380467772.9800	5.3172
26_District1	278	1000	47978959669600.5700	42180173025536.8000	13.7476
26_District2	42	1000	330365825046.6699	282951685653.1000	16.7569
26_District3	313	1000	1116818510591.6294	1171533328289.0400	4.8991 *
26_District4	888	1000	120236047561561.1900	105580668249051.0000	13.8807
26_District5	983	1000	86502277247398.5300	76363123397244.6000	13.2775
27_District0	11	1000	161902499001.3600	162479691641.3000	0.3565 *
27_District1	227	1000	21716688296489.9840	18425601260045.1000	17.8614
27_District2	73	1000	81122339457.5800	61192189969.0700	32.5697
27_District3	73	1000	124415563572.2199	116712931257.8400	6.5996
27_District4	629	1000	42327497051910.4000	33778988955119.3000	25.3071
27_District5	760	1000	41801483784076.4700	39568513214098.5000	5.6433
28_District0	11	1000	113195650438.2499	102108349093.8000	10.8583
28_District1	367	1000	15869647127759.2970	15128335139531.3000	4.9001
28_District2	32	1000	75831813269.1900	64692968024.3600	17.2180
28_District3	110	1000	788663590439.0907	805624097687.0900	2.1505 *
28_District4	678	1000	27419587122935.0040	22982825117023.6000	19.3046
28_District5	392	1000	37494457433440.9500	33791534896023.1000	10.9581
29_District0	27	1000	32047331066.6800	29353316624.4500	9.1778
29_District1	87	1000	8026953178617.5590	7484826255380.2500	7.2430
29_District2	85	1000	98239088277.6600	101233312043.5700	3.0478 *
29_District3	120	1000	227190299236.9698	232601979076.2900	2.3820 *
29_District4	516	1000	9873197805012.6880	8309110124637.3200	18.8237
29_District5	567	1000	37288916047738.3400	35670467010142.6000	4.5372
2_District0	23	1000	165898766697.2700	147005506881.3000	12.8520
2_District1	552	1000	44070915248314.9100	39540340925649.8000	11.4581
2_District2	70	1000	230500651640.3700	217829855689.6200	5.8168
2_District3	56	1000	917964591587.2301	858787147589.8100	6.8908
2_District4	1215	1000	70464164182588.3900	58876326066126.1000	19.6816
2_District5	529	1000	111088965128385.5800	95903342380824.9000	15.8342
30_District0	9	1000	16973733842.6800	14856579555.0000	14.2506
30_District1	125	1000	4662589808304.7400	4288991390631.0100	8.7106
30_District2	33	1000	42672726517.7900	28984654012.0000	47.2252
30_District3	60	1000	142654059134.0700	140260859804.8700	1.7062
30_District4	340	1000	6587179976482.2970	6033193692111.0700	9.1823
30_District5	175	1000	5740311807405.0600	5491783393822.4800	4.5254
3_District0	11	1000	97925708583.0300	90701681116.3600	7.9646
3_District1	146	1000	18776108633160.5800	17931112587450.7000	4.7124
3_District2	66	1000	196338744927.1999	184423936337.4900	6.4605
3_District3	185	1000	848011293090.7697	837894240881.4400	1.2074
3_District4	1433	1000	83315086685172.5200	67042838268903.4000	24.2714
3_District5	725	1000	67954724342790.4200	60566955554266.6000	12.1976
4_District0	8	1000	22005064052.4900	20379141775.5100	7.9783
4_District1	330	1000	17765728314854.9340	16091560843817.3000	10.4040
4_District2	34	1000	20924946330.8200	19771989271.8700	5.8312

Table E.2 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
4_District3	52	1000	31495961873.6600	31243041764.1500	0.8095
4_District4	1425	1000	52583890084638.5100	44497725128783.0000	18.1720
4_District5	286	1000	34337750896834.4000	30292083288905.3000	13.3555
5_District0	31	1000	105380059751.4399	105305055950.1000	0.0712
5_District1	248	1000	25117238052545.0200	22578812619408.6000	11.2425
5_District2	37	1000	77176335809.1900	49746631100.9200	55.1388
5_District3	109	1000	325134601803.4000	366270241695.4900	12.6518 *
5_District4	676	1000	79770967975230.9700	68243049339872.1000	16.8924
5_District5	377	1000	129624840984169.8000	121803986176569.0000	6.4208
6_District0	11	1000	190513021210.9900	174024536017.0000	9.4748
6_District1	455	1000	27967874791714.0160	24635464198083.6000	13.5268
6_District2	120	1000	260097530699.0004	263920214770.7000	1.4697 *
6_District3	103	1000	296599575499.1701	272026860685.1900	9.0331
6_District4	622	1000	38148233016413.3800	32671893526336.1000	16.7616
6_District5	439	1000	60567642095460.7500	52577668995018.6000	15.1965
7_District0	37	1000	52751552955.0599	53992400830.1800	2.3522 *
7_District1	681	1000	28123757784067.8160	25530776489059.8000	10.1562
7_District2	15	1000	109565973962.9600	96812988604.0900	13.1728
7_District3	74	1000	671490236173.9000	627921570619.9200	6.9385
7_District4	468	1000	35016845678279.5080	30888008083772.9000	13.3671
7_District5	671	1000	58969299841531.7100	52214576657422.1000	12.9364
8_District0	9	1000	57495503324.4900	47815962594.9100	20.2433
8_District1	249	1000	14549666982668.8600	13654929607346.6000	6.5524
8_District2	55	1000	76153212924.0699	73150043166.7900	4.1054
8_District3	93	1000	370838508187.7199	367102727288.8900	1.0176
8_District4	457	1000	32056205765809.1640	29816748603227.9000	7.5107
8_District5	264	1000	44577398258198.5550	40450402089088.4000	10.2026
9_District0	8	1000	109160592872.5800	100093204504.3000	9.0589
9_District1	144	1000	13341592005296.2360	12178217016824.2000	9.5529
9_District2	22	1000	115662514136.1400	89342594098.5600	29.4595
9_District3	148	1000	1164159893069.3293	911115793606.4100	27.7729
9_District4	805	1000	45475409704757.2200	37164202673658.1000	22.3634
9_District5	253	1000	29855182828065.2800	25382936280280.4000	17.6191
C101_100t_20w	123	1000	48319872397.5998	-12304901708.1200	492.6879
C101_25t_5w	6	1000	111640277.2000	9512566.6400	1073.6083
C101_50t_10w	19	1000	755802496.6800	-1079154704.1200	170.0365
C102_100t_20w	259	1000	62399995381.8000	-5155408825.8200	1310.3791
C102_25t_5w	9	1000	78098258.9200	-37546115.7600	308.0062
C102_50t_10w	55	1000	1437580703.2999	-1102530180.6200	230.3892
C103_100t_20w	759	1000	39930163004.1205	-7222438475.8600	652.8626
C103_25t_5w	31	1000	76596467.7199	-43052838.4600	277.9127
C103_50t_10w	141	1000	2551801064.3399	-1110321792.2400	329.8253
C104_100t_20w	1036	1000	17314423586.3605	-8681519205.2400	299.4400
C104_25t_5w	48	1000	36546720.0199	-33540706.8600	208.9622
C104_50t_10w	367	1000	155837939.9399	-1086946666.3200	114.3372
C105_100t_20w	243	1000	25801404717.2202	-12158993843.3000	312.2001
C105_25t_5w	6	1000	123154424.4199	1002190.7200	12188.5217
C105_50t_10w	40	1000	-342833993.7200	-1102529614.6600	68.9047
C106_100t_20w	368	1000	25582543085.0799	-5349954009.7600	578.1824
C106_25t_5w	9	1000	116646382.1600	9512566.6400	1126.2345
C106_50t_10w	38	1000	1363558947.8400	-1086946589.7000	225.4485
C107_100t_20w	422	1000	39857208909.1199	-6979258964.6200	671.0808
C107_25t_5w	9	1000	76596508.9400	-2001420.3200	3927.1075
C107_50t_10w	80	1000	163629531.8600	-1125905161.9000	114.5331
C108_100t_20w	561	1000	25096183365.7199	-6492899831.1000	486.5173
C108_25t_5w	14	1000	113142101.6399	-8009072.9800	1512.6741
C108_50t_10w	111	1000	-412959808.7800	-1125905177.3200	63.3219
C109_100t_20w	862	1000	25315044799.9998	-6614488230.8200	482.7211
C109_25t_5w	19	1000	74093372.6799	-26031566.9600	384.6289
C109_50t_10w	126	1000	-444127049.5800	-1125905761.1800	60.5537
C201_100t_20w	980	1000	-12596716053.9600	-12596720162.2300	0.0000
C201_25t_5w	38	1000	-43553212.8200	-45555916.9200	4.3961
C201_50t_10w	189	1000	-1102529455.1600	-1125905241.3800	2.0761
C202_100t_20w	2133	1000	24488233975.9598	-6055175633.1600	504.4182
C202_25t_5w	256	1000	-44554590.5800	-45555950.8800	2.1980
C202_50t_10w	529	1000	89608214.7199	-1125905361.9600	107.9587
C203_100t_20w	3102	1000	2334535771.0799	-6930623788.8800	133.6843
C203_25t_5w	1274	1000	-42552020.9000	-45555956.2400	6.5939
C203_50t_10w	1678	1000	120775326.5199	-1125905456.0600	110.7269
C204_100t_20w	6053	1000	9848798303.7597	-8657198950.2400	213.7642
C204_25t_5w	1472	1000	-41050214.7200	-45555977.0000	9.8906
C204_50t_10w	1970	1000	-483084988.7400	-1125905585.1400	57.0936
C205_100t_20w	1945	1000	9702889586.1598	-12596718687.8400	177.0271
C205_25t_5w	90	1000	-42552102.6999	-45555926.8000	6.5937
C205_50t_10w	550	1000	-1118112648.1400	-1125905403.9800	0.6921
C206_100t_20w	3312	1000	17119879659.0198	-7733117210.9600	321.3839
C206_25t_5w	139	1000	-41050178.5199	-45555926.8000	9.8905
C206_50t_10w	906	1000	-514252208.6600	-1125905541.4200	54.3254
C207_100t_20w	4995	1000	2285899598.6398	-7222438653.2400	131.6499
C207_25t_5w	126	1000	-41050287.9800	-45555933.4200	9.8903
C207_50t_10w	2134	1000	-522043906.6600	-1125905455.9000	53.6334
C208_100t_20w	4142	1000	17168515513.8997	-8657201177.3800	298.3148
C208_25t_5w	205	1000	-44053965.5799	-45555933.1600	3.2969
C208_50t_10w	1216	1000	101295859.4999	-1125905465.6600	108.9968
hh_00_P0	167	1000	597320465109.3570	6663651008.2900	8863.8617
ll1_00_P0	44	1000	8730580967.6700	1338732826.5000	552.1526
ll1_01_P0	42	1000	8761695462.9999	1338732826.5000	554.4767
ll1_02_P0	41	1000	8761705984.6401	1338732826.5000	554.4775
ll1_03_P0	41	1000	8668351291.4799	1338732826.5000	547.5042
ll1_04_P0	43	1000	10333442162.1100	1338732826.5000	671.8823
ll1_05_P0	36	1000	5509254527.9397	1307660206.1200	321.3062
ll1_06_P0	42	1000	12029720714.7801	1214281948.4400	890.6859
ll1_07_P0	46	1000	5400270252.8000	1338732826.5000	303.3867
ll2_00_P0	31	1000	-160646628.2000	85352916.2700	153.1308 *
ll3_00_P0	35	1000	393003407.4499	-198615845.6200	297.8711
R101_100t_20w	8	1000	19386853826.0999	6319981502.8600	206.7549
R101_25t_5w	2	1000	60631893.5799	47893886.5800	26.5963

Table E.2 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
R101_50t_10w	4	1000	1303821345.7999	709051142.3992	83.8825
R102_100t_20w	14	1000	19284177804.6200	14101742003.1800	36.7503
R102_25t_5w	2	1000	38715883.0199	34655177.0000	11.7174
R102_50t_10w	4	1000	1035439003.2199	641955625.9388	61.2944
R103_100t_20w	22	1000	15263601941.3799	12469734030.7400	22.4051
R103_25t_5w	3	1000	34098884.1000	30260803.3400	12.6833
R103_50t_10w	8	1000	778744364.3399	570531480.7600	36.4945
R104_100t_20w	35	1000	11140350355.3797	14023383931.9600	25.8792 *
R104_25t_5w	3	1000	30260846.9599	29982663.2800	0.9278
R104_50t_10w	13	1000	633298589.4200	293491458.8800	115.7809
R105_100t_20w	16	1000	16882099731.7799	8424839701.0800	100.3848
R105_25t_5w	3	1000	47782625.1400	35155829.6400	35.9166
R105_50t_10w	4	1000	967045008.0399	388290958.0600	149.0516
R106_100t_20w	19	1000	17649468147.7000	8443754034.7200	109.0239
R106_25t_5w	3	1000	34822056.7200	30483330.4600	14.2331
R106_50t_10w	5	1000	699961106.3599	303880567.4400	130.3408
R107_100t_20w	33	1000	13593766313.9999	11634816307.3400	16.8369
R107_25t_5w	3	1000	34265792.3800	22028382.6800	55.5529
R107_50t_10w	9	1000	768355395.5599	372274723.4800	106.3947
R108_100t_20w	41	1000	11086310338.7398	12369760088.0800	11.5768 *
R108_25t_5w	3	1000	26255816.5000	26033406.5200	0.8543
R108_50t_10w	11	1000	636761391.6200	293491505.9600	116.9607
R109_100t_20w	26	1000	15244687856.7399	6636116061.1200	129.7230
R109_25t_5w	3	1000	46781479.3800	26255954.0600	78.1747
R109_50t_10w	6	1000	702125519.8599	631133985.4200	11.2482
R110_100t_20w	25	1000	13550534201.0599	7622345982.8200	77.7738
R110_25t_5w	2	1000	38882734.4399	34488424.6600	12.7414
R110_50t_10w	6	1000	702991274.6800	636761297.5800	10.4010
R111_100t_20w	29	1000	15263602173.7799	8473476077.2600	80.1338
R111_25t_5w	3	1000	30260922.0400	26144634.7600	15.7442
R111_50t_10w	6	1000	510362057.0400	496942686.1600	2.7003
R112_100t_20w	37	1000	13674825975.4399	10851236130.0600	26.0209
R112_25t_5w	3	1000	34377155.8199	26311467.9000	30.6546
R112_50t_10w	7	1000	569232951.9199	427249792.1800	33.2318
R201_100t_20w	369	1000	-570115529.6601	-1383419122.5000	58.7893
R201_25t_5w	15	1000	-5060130.4600	-5171519.4600	2.1538
R201_50t_10w	135	1000	-124231486.3201	-125097475.6400	0.6922
R202_100t_20w	368	1000	2726325051.0198	-778169509.0800	450.3510
R202_25t_5w	27	1000	-721406.9600	-5171550.1800	86.0504
R202_50t_10w	109	1000	78786732.2199	-125097689.9000	162.9801
R203_100t_20w	445	1000	1902214182.8599	-891653653.5400	313.3355
R203_25t_5w	42	1000	-443367.6600	-5171614.5200	91.4269
R203_50t_10w	370	1000	11690997.3600	-125097805.5400	109.3454
R204_100t_20w	922	1000	1097018656.7198	-1007840112.0800	208.8484
R204_25t_5w	51	1000	-387769.5999	-5060450.1000	92.3372
R204_50t_10w	367	1000	12556892.2799	-125098025.1800	110.0376
R205_100t_20w	500	1000	-1356397288.1200	-910567724.8000	32.8686 *
R205_25t_5w	27	1000	-721586.6599	-5171759.2000	86.0475
R205_50t_10w	234	1000	-58867560.4000	-125097876.3800	52.9427
R206_100t_20w	646	1000	4382650938.2199	-964607487.1800	554.3455
R206_25t_5w	42	1000	-4726239.9600	-5171813.9000	8.6154
R206_50t_10w	182	1000	9093533.6599	-125097837.4400	107.2691
R207_100t_20w	678	1000	1926532415.4799	-1018647805.1000	289.1264
R207_25t_5w	62	1000	-5004507.5999	-5171794.9600	3.2346
R207_50t_10w	280	1000	75323806.6199	-125097967.7600	160.2118
R208_100t_20w	931	1000	-1356397427.0201	-1042965700.1000	23.1076 *
R208_25t_5w	55	1000	-4782173.0600	-5060560.5800	5.5011
R208_50t_10w	726	1000	9093863.3199	-125098091.9200	107.2693
R209_100t_20w	682	1000	286418854.4199	-1013243421.1000	128.2675
R209_25t_5w	32	1000	-443366.7000	-5060483.0600	91.2386
R209_50t_10w	260	1000	-123365830.7000	-125097999.8800	1.3846
R210_100t_20w	622	1000	2737132871.9199	-978117773.2200	379.8367
R210_25t_5w	54	1000	-554675.4799	-5171642.3200	89.2746
R210_50t_10w	274	1000	77055164.6999	-125097934.6800	161.5958
R211_100t_20w	785	1000	6033572354.8598	-942991911.9800	739.8328
R211_25t_5w	46	1000	-443441.7800	-5060502.8400	91.2371
R211_50t_10w	240	1000	-51508879.8200	-122500836.0400	57.9522
RC101_100t_20w	11	1000	20262302139.0999	15814809991.8800	28.1223
RC101_25t_5w	3	1000	39438835.2199	27034405.3800	45.8838
RC101_50t_10w	4	1000	784371669.9399	190034189.6200	312.7529
RC102_100t_20w	13	1000	16147156689.9798	15822916717.4800	2.0491
RC102_25t_5w	3	1000	43054541.1400	26645148.4800	61.5849
RC102_50t_10w	4	1000	842810143.6400	578756222.9600	45.6243
RC103_100t_20w	19	1000	17700806751.2397	17527878844.9600	0.9865
RC103_25t_5w	3	1000	26311540.7799	21805915.8800	20.6623
RC103_50t_10w	6	1000	573994460.1400	767922882.8400	33.7857 *
RC104_100t_20w	27	1000	12023904712.8598	18130424572.1200	50.7864 *
RC104_25t_5w	4	1000	29815859.0399	25810879.2400	15.5166
RC104_50t_10w	7	1000	567501470.5999	498674346.2200	13.8020
RC105_100t_20w	12	1000	21043180413.7999	10907978098.4200	92.9154
RC105_25t_5w	3	1000	43165777.4599	30761388.4200	40.3245
RC105_50t_10w	5	1000	1041499512.4200	448893536.2400	132.0148
RC106_100t_20w	20	1000	17011796530.5399	11529438265.0600	47.5509
RC106_25t_5w	3	1000	39438907.6600	22417682.5200	75.9276
RC106_50t_10w	5	1000	780475953.4799	515989390.5800	51.2581
RC107_100t_20w	20	1000	16222812485.9399	16638920438.6600	2.5649 *
RC107_25t_5w	3	1000	35267047.6000	30761589.8800	14.6463
RC107_50t_10w	6	1000	579189146.1400	575726019.2200	0.6015
RC108_100t_20w	25	1000	13685634437.0399	14823176358.6800	8.3119 *
RC108_25t_5w	3	1000	35267109.7599	30372205.5600	16.1163
RC108_50t_10w	6	1000	581786519.9200	630268451.2000	8.3332 *
RC201_100t_20w	339	1000	-1388819673.9802	-1378013035.7800	0.7781 *
RC201_25t_5w	12	1000	-4447346.8800	-5171005.4000	13.9945
RC201_50t_10w	79	1000	-122498977.5600	-125096456.0400	2.0763
RC202_100t_20w	395	1000	1937341330.3399	-848419159.5200	328.3471
RC202_25t_5w	16	1000	-4892928.1600	-5171480.4200	5.3863
RC202_50t_10w	79	1000	77921655.9399	-125096955.0600	162.2890

Table E.2 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
RC203_100t_20w	369	1000	1094317604.3599	-915968950.1800	219.4710
RC203_25t_5w	40	1000	-721409.5599	-5171505.3800	86.0502
RC203_50t_10w	159	1000	77055884.3598	-125096765.1800	161.5970
RC204_100t_20w	533	1000	-532286353.4402	-1032155821.9000	48.4296
RC204_25t_5w	78	1000	-4781581.5600	-5060413.3600	5.5100
RC204_50t_10w	382	1000	10826253.1998	-125097014.9000	108.6542
RC205_100t_20w	396	1000	1083509820.6998	-856525543.8000	226.5005
RC205_25t_5w	18	1000	-442913.0600	-5171462.0600	91.4354
RC205_50t_10w	99	1000	-48910172.8600	-125096766.4200	60.9021
RC206_100t_20w	436	1000	-564709306.6801	-905161565.1000	37.6123
RC206_25t_5w	21	1000	-4726111.6600	-5171397.9400	8.6105
RC206_50t_10w	158	1000	-122932272.4599	-125097137.7600	1.7305
RC207_100t_20w	464	1000	1910321553.7198	-907864881.6600	310.4191
RC207_25t_5w	28	1000	-4336779.3200	-5059890.2000	14.2910
RC207_50t_10w	146	1000	-120767762.7200	-122499526.2000	1.4136
RC208_100t_20w	580	1000	2783068026.3798	-978116228.2000	384.5334
RC208_25t_5w	45	1000	-4726621.3600	-5060744.0000	6.6022
RC208_50t_10w	201	1000	-54970815.0800	-125096819.1800	56.0573
test150-0-0-0-0_d0_tw0	2163	1000	101050622678.1997	-28349336446.9000	456.4479
test150-0-0-0-0_d0_tw1	759	1000	-24141770230.7996	-30832491493.7000	21.7002
test150-0-0-0-0_d0_tw2	565	1000	7242547420.0989	-28832172176.8000	125.1196
test150-0-0-0-0_d0_tw3	441	1000	6897666282.2002	-27383664735.0000	125.1889
test150-0-0-0-0_d0_tw4	508	1000	163819258695.0000	-26142085667.0000	726.6495
test250-0-0-0-0_d0_tw0	-	-	-	23650649218.5000	-
test250-0-0-0-0_d0_tw1	1518	1000	40543948170.6739	-292254131472.0000	113.8728
test250-0-0-0-0_d0_tw2	1291	1000	550721722328.1854	-275698693917.1000	299.7549
test250-0-0-0-0_d0_tw3	254	249	-214544933324.4147	-266914175565.0000	19.6202
test250-0-0-0-0_d0_tw4	1568	1000	1571077267390.7969	-231776103927.7000	777.8426
test50-0-0-0-0_d0_tw0	320	812	-842599013.7000	-842599324.2000	0.0000
test50-0-0-0-0_d0_tw1	150	1000	-842598772.9000	-842599506.6000	0.0000
test50-0-0-0-0_d0_tw2	119	1000	-842598460.5000	-842599560.3000	0.0001
test50-0-0-0-0_d0_tw3	89	1000	-836356323.7000	-842598590.0000	0.7408
test50-0-0-0-0_d0_tw4	101	1000	-842598554.5999	-842599753.8000	0.0001

E.3 Tabu Search Results: Config. 3

Table E.3: Tabu Search experiments results with parameter configuration 3

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
10_District0	932	100000	61319422104.5999	71563566909.3600	16.7061 *
10_District1	6115	100000	8146632239585.1520	8218857439409.0600	0.8865 *
10_District2	4339	100000	40104783192.0499	27771011394.0200	44.4123
10_District3	1903	100000	124043366704.6098	139155792160.4600	12.1831 *
10_District4	7200	45488	24044906203637.3630	21291228411410.1000	12.9333
10_District5	7200	66731	39385723841178.7660	37249297856014.8000	5.7354
11_District0	1079	100000	3394771765.6100	3046672128.3800	11.4255
11_District1	7200	68177	6315634612575.4090	6157994611789.7000	2.5599
11_District2	1969	100000	12633316702.5300	9214556192.4000	37.1017
11_District3	1418	100000	58081338825.5601	61007292651.7800	5.0376 *
11_District4	7200	28498	24626573447061.3300	22460659385239.4000	9.6431
11_District5	7200	67174	15979414299649.3100	15643005576898.3000	2.1505
12_District0	1687	100000	123344728753.0399	115036288619.9000	7.2224
12_District1	7201	51425	40934843838070.0900	37877356021322.0000	8.0720
12_District2	2670	100000	148865845239.4801	153054550706.1000	2.8137 *
12_District3	4638	100000	213304502089.9999	239836499336.7000	12.4385 *
12_District4	7200	50883	60812594795951.9300	59129219661289.8000	2.8469
12_District5	7200	35075	68537201389243.0200	65945373929847.0000	3.9302
13_District0	452	100000	167512795988.1300	154315121626.0000	8.5524
13_District1	7200	75584	16738317958496.3160	14674175609787.1000	14.0664
13_District2	2683	100000	108881728883.8301	126837068235.0100	16.4906 *
13_District3	7200	85817	404206934379.0810	429665639491.2800	6.2984 *
13_District4	7200	87122	30666882573676.4960	30033656135618.7000	2.1083
13_District5	4825	100000	45852147317397.7100	42764648834839.3000	7.2197
14_District0	587	100000	35108024849.6400	34977146773.4600	0.3741
14_District1	7200	54994	12327536190349.4700	12434388027267.4000	0.8667 *
14_District2	1238	100000	88582195166.0800	90165619840.7900	1.7875 *
14_District3	6610	100000	231997649363.2798	262406322982.9800	13.1073 *
14_District4	7200	54277	37546192404006.6640	31546738861338.4000	19.0176
14_District5	7200	85338	44223833356315.5300	44524141233501.8000	0.6790 *
15_District0	442	100000	56800250389.3900	42188641727.2300	34.6339
15_District1	7200	94308	12423010816403.0400	12317798430422.9000	0.8541
15_District2	955	100000	76781032874.0000	67421894826.9300	13.8814
15_District3	3570	100000	371671849322.2302	463823619193.1600	24.7938 *
15_District4	7200	44524	27005258718010.6500	22834329911998.4000	18.2660
15_District5	7200	61769	29665152298463.5600	28700034523627.8000	3.3627
16_District0	827	100000	123337798475.7000	120807470457.0500	2.0945
16_District1	7200	99573	13406517183821.4160	12316160134202.9000	8.8530
16_District2	1622	100000	93711639337.6000	94504646913.7700	0.8462 *
16_District3	4736	100000	193869441381.4698	210518442436.7200	8.5877 *
16_District4	7200	56491	31188308228108.7270	28327769996973.5000	10.0980
16_District5	7200	78191	48288514170261.2700	48522145053303.8000	0.4838 *
17_District0	484	100000	66543877940.0502	60633779564.4100	9.7472
17_District1	7200	60593	11613753631516.3360	12050832937058.4000	3.7634 *
17_District2	2732	100000	110334481168.8099	111787046906.6000	1.3165 *
17_District3	1208	100000	180960705402.5399	178730940805.7800	1.2475
17_District4	7200	57304	20255890937012.8000	19092974442252.1000	6.0908
17_District5	5947	100000	24371976202314.5740	23886759047749.6000	2.0313

Table E.3 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
18_District0	185	100000	2860280110.3899	2341527488.6000	22.1544
18_District1	4264	100000	13641483090523.8070	13924688690304.7000	2.0760 *
18_District2	2094	100000	6296812342.2499	4417358464.4100	42.5470
18_District3	807	100000	55310433032.8400	70920528499.0700	28.2226 *
18_District4	7200	56855	17713764545605.6900	18718722507656.5000	5.6733 *
18_District5	5948	100000	12763728334951.8900	12641617614178.6000	0.9659
19_District0	613	100000	45460485793.5800	55462969817.4300	22.0025 *
19_District1	7200	76567	22852643800475.0700	22401449014876.9000	2.0141
19_District2	2447	100000	146230091895.3399	153280470262.9000	4.8214 *
19_District3	2737	100000	209128517787.9201	243569037679.2000	16.4685 *
19_District4	7200	48894	114019612823637.7500	106192108824987.0000	7.3710
19_District5	7200	76320	83619359479325.6400	78807991381031.8000	6.1051
1_District0	703	100000	414857520738.6203	430590936121.1900	3.7924 *
1_District1	7200	53296	54575711701779.8200	56270206789859.1000	3.1048 *
1_District2	1896	100000	621503655933.5693	672001346398.5800	8.1250 *
1_District3	7200	63653	1012224604297.2406	1031045184732.7300	1.8593 *
1_District4	7200	33475	74279646462691.5500	73732177837593.4000	0.7425
1_District5	7200	62952	90169588524306.8100	91806570717137.9000	1.8154 *
20_District0	594	100000	92857415991.1800	98421186256.3800	5.9917 *
20_District1	7200	79309	17810569182545.7930	17222315357479.6000	3.4156
20_District2	3370	100000	29604771245.8499	27378465012.0200	8.1315
20_District3	7200	77701	274089797172.5999	305213925793.8100	11.3554 *
20_District4	7200	50587	26818953058689.6680	24960379220572.1000	7.4460
20_District5	7200	58011	47470304190301.7000	48965990159239.8000	3.1507 *
21_District0	3209	100000	103690074086.5299	120414278382.3300	16.1290 *
21_District1	7200	90372	16992532633164.9980	16893868904739.4000	0.5840
21_District2	4715	100000	76409280808.5299	62666013838.2000	21.9309
21_District3	4615	100000	818077435822.7802	892017226174.9800	9.0382 *
21_District4	7200	58441	24138789734561.3200	24304758516183.0000	0.6875 *
21_District5	7200	84118	47370400960879.9100	45930297567746.2000	3.1354
22_District0	334	100000	146677189834.1199	138906549645.3000	5.5941
22_District1	3361	100000	15075208860193.2710	14071948171020.4000	7.1295
22_District2	1522	100000	241645162667.8598	226837888943.8600	6.5276
22_District3	6807	100000	290127101612.7802	285892462254.7700	1.4812
22_District4	7200	59862	30953145205340.6200	29074924558505.3000	6.4599
22_District5	7200	76447	41928073820889.2400	42772519577575.2000	2.0140 *
23_District0	513	100000	35383951512.4099	32906581132.3800	7.5284
23_District1	7200	61698	17282011527164.6640	17211576588480.4000	0.4092
23_District2	925	100000	198436996077.3700	183871311138.5000	7.9216
23_District3	4363	100000	194216366208.3199	250619333021.8100	29.0413 *
23_District4	7200	56413	35302677247038.2000	34555727151866.1000	2.1615
23_District5	7200	63861	73432793244856.6900	70608343796642.5000	4.0001
24_District0	1096	100000	120480244542.2399	105830236823.5000	13.8429
24_District1	7200	90127	12951062920382.2600	11389191027676.9000	13.7136
24_District2	3213	100000	23750977920.4000	26055867878.7800	9.7044 *
24_District3	6953	100000	101722806632.5799	159315579521.2200	56.6173 *
24_District4	7200	57921	24698884569514.2580	24517137738123.8000	0.7413
24_District5	7200	96573	26574525301707.4600	24738655441667.6000	7.4210
25_District0	470	100000	1454217691.2299	1255504790.1000	15.8273
25_District1	7200	51467	5606702280836.2930	6071232244353.8300	8.2852 *
25_District2	708	100000	2756992655.7599	2430507030.7800	13.4328
25_District3	1601	100000	44718984059.7299	54825213021.3900	22.5994 *
25_District4	7200	54241	22300242439913.5550	20503174454937.0000	8.7648
25_District5	6809	100000	10775069860093.8100	11011719621667.4000	2.1962 *
26_District0	1335	100000	209058673246.5900	225380467772.9800	7.8072 *
26_District1	7200	85142	43439619436478.1300	42180173025536.8000	2.9858
26_District2	1819	100000	330309041227.7099	282951685653.1000	16.7369
26_District3	7200	42382	973542034605.6295	1171533328289.0400	20.3372 *
26_District4	7200	22970	114403937232307.2800	105580668249051.0000	8.3568
26_District5	7200	57763	75887350198066.8300	76363123397244.6000	0.6269 *
27_District0	998	100000	146087424745.8100	162479691641.3000	11.2208 *
27_District1	7200	47325	20423182676689.6250	18425601260045.1000	10.8413
27_District2	2006	100000	64148580288.2600	61192189969.0700	4.8313
27_District3	5242	100000	118518236254.6900	116712931257.8400	1.5467
27_District4	7201	68189	36106798229254.8900	33778988955119.3000	6.8912
27_District5	7200	59743	39813786783293.8200	39568513214098.5000	0.6198
28_District0	501	100000	120470321903.0599	102108349093.8000	17.9828
28_District1	7200	74949	14317926873085.4200	15128335139531.3000	5.6600 *
28_District2	1743	100000	72015727315.7099	64692968024.3600	11.3192
28_District3	6290	100000	738620775738.5497	805624097687.0900	9.0714 *
28_District4	7200	47047	23907581610881.0200	22982825117023.6000	4.0236
28_District5	7200	73029	35754345308886.6400	33791534896023.1000	5.8085
29_District0	2101	100000	31974520148.4099	29353316624.4500	8.9298
29_District1	3082	100000	7288752882344.6420	7484826255380.2500	2.6900 *
29_District2	2366	100000	83239449171.6500	101233312043.5700	21.6169 *
29_District3	4990	100000	194319361708.5997	232601979076.2900	19.7008 *
29_District4	7200	78791	8872604416378.6900	8309110124637.3200	6.7816
29_District5	7200	30804	36360343783905.1800	35670467010142.6000	1.9340
2_District0	1150	100000	132743550630.9400	147005506881.3000	10.7439 *
2_District1	7200	70375	40767511066836.0000	39540340925649.8000	3.1035
2_District2	3362	100000	185793411606.8301	217829855689.6200	17.2430 *
2_District3	2804	100000	813179704074.5292	858787147589.8100	5.6085 *
2_District4	7200	18577	63701520811872.0700	58876326066126.1000	8.1954
2_District5	7200	72740	102748864012452.6900	95903342380824.9000	7.1379
30_District0	467	100000	14923501548.0199	14856579555.0000	0.4504
30_District1	6868	100000	4420744720941.4550	4288991390631.0100	3.0718
30_District2	2240	100000	37973365004.3800	28984654012.0000	31.0119
30_District3	4204	100000	118851432884.6800	140260859804.8700	18.0136 *
30_District4	7200	78605	5845835802053.5860	6033193692111.0700	3.2049 *
30_District5	4286	100000	5338680228691.9880	5491783393822.4800	2.8678 *
3_District0	461	100000	97493501748.1100	90701681116.3600	7.4880
3_District1	7122	100000	17564651144844.1720	17931112587450.7000	2.0863 *
3_District2	2083	100000	166532062526.5700	184423936337.4900	10.7438 *
3_District3	7200	87179	768491265231.4299	837894240881.4400	9.0310 *
3_District4	7200	24573	73102229718603.8300	67042838268903.4000	9.0380
3_District5	7200	34792	64452705212649.9900	60566955554266.6000	6.4156
4_District0	206	100000	21102539332.5700	20379141775.5100	3.5496

Table E.3 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
4_District1	7200	82420	16222467374387.1170	16091560843817.3000	0.8135
4_District2	1121	100000	21157990626.6300	19771989271.8700	7.0099
4_District3	1701	100000	31243042239.1799	31243041764.1500	0.0000
4_District4	7200	24660	47365396000871.1640	44497725128783.0000	6.4445
4_District5	7200	66652	30518149884277.8700	30292083288905.3000	0.7462
5_District0	2296	100000	105398810657.5599	105305055950.1000	0.0890
5_District1	7200	66325	22650347487648.3750	22578812619408.6000	0.3168
5_District2	2105	100000	72863184346.2799	49746631100.9200	46.4685
5_District3	4939	100000	283434600221.7600	366270241695.4900	29.2256 *
5_District4	7200	38172	77339074381850.3800	68243049339872.1000	13.3288
5_District5	7200	68881	123530120370460.6200	121803986176569.0000	1.4171
6_District0	641	100000	184399313770.2801	174024536017.0000	5.9616
6_District1	7200	97582	24054291791237.3160	24635464198083.6000	2.4160 *
6_District2	2159	100000	238814475647.1300	263920214770.7000	10.5126 *
6_District3	3832	100000	267141923694.0102	272026860685.1900	1.8285 *
6_District4	7200	71215	32875720575170.0500	32671893526336.1000	0.6238
6_District5	7200	59196	56074931312460.6300	52577668995018.6000	6.6516
7_District0	2194	100000	48036330914.6500	53992400830.1800	12.3990 *
7_District1	7200	69683	25334281259296.4400	25530776489059.8000	0.7756 *
7_District2	1372	100000	104190522394.5999	96812988604.0900	7.6203
7_District3	3377	100000	545854362657.1897	627921570619.9200	15.0346 *
7_District4	7200	59246	32370401586623.4300	30888008083772.9000	4.7992
7_District5	7200	53860	51257539533425.8050	52214576657422.1000	1.8671 *
8_District0	326	100000	52728450712.1200	47815962594.9100	10.2737
8_District1	7200	70411	14140270566964.2290	13654929607346.6000	3.5543
8_District2	2940	100000	66805600666.7400	73150043166.7900	9.4968 *
8_District3	6351	100000	342259784786.4702	367102727288.8900	7.2585 *
8_District4	7200	34767	29568198985619.5270	29816748603227.9000	0.8405 *
8_District5	6850	100000	42595956819177.9800	40450402089088.4000	5.3041
9_District0	344	100000	102598122380.8400	100093204504.3000	2.5025
9_District1	5035	100000	11931035621717.5350	12178217016824.2000	2.0717 *
9_District2	1150	100000	102349044390.6600	89342594098.5600	14.5579
9_District3	7200	85876	954523624499.5204	911115793606.4100	4.7642
9_District4	7200	39788	39783518017645.9600	37164202673658.1000	7.0479
9_District5	6231	100000	28411688890197.5860	25382936280280.4000	11.9322
C101_100t_20w	7102	100000	-10091963997.6196	-12304901708.1200	17.9841
C101_25t_5w	232	100000	114643937.6200	9512566.6400	1105.1840
C101_50t_10w	1244	100000	-919424005.6801	-1079154704.1200	14.8014
C102_100t_20w	7200	39812	26603898612.6599	-5155408825.8200	616.0385
C102_25t_5w	1022	100000	39049509.9801	-37546115.7600	204.0041
C102_50t_10w	3769	100000	-424647731.5605	-1102530180.6200	61.4842
C103_100t_20w	7200	26897	18457369523.0197	-7222438475.8600	355.5559
C103_25t_5w	3094	100000	-1000307.7797	-43052838.4600	97.6765
C103_50t_10w	7200	58781	790865758.1398	-1110321792.2400	171.2285
C104_100t_20w	7200	15802	2869531113.5005	-8681519205.2400	133.0533
C104_25t_5w	3780	100000	43555143.7399	-33540706.8600	229.8575
C104_50t_10w	7200	22901	-490877464.6000	-1086946666.3200	54.8388
C105_100t_20w	7200	53509	-10748548922.1799	-12158993843.3000	11.6000
C105_25t_5w	351	100000	154193281.5800	1002190.7200	15285.6225
C105_50t_10w	2591	100000	-409063950.8399	-1102529614.6600	62.8976
C106_100t_20w	7200	29978	-10651277134.4797	-5349954009.7600	49.7717 *
C106_25t_5w	328	100000	82103177.3201	9512566.6400	763.1022
C106_50t_10w	2032	100000	-405168413.3997	-1086946589.7000	62.7241
C107_100t_20w	7200	34654	-10845821722.5401	-6979258964.6200	35.6502 *
C107_25t_5w	447	100000	149187091.3000	-2001420.3200	7554.0609
C107_50t_10w	5046	100000	-970069980.6800	-1125905161.9000	13.8408
C108_100t_20w	7200	27943	2942485491.1397	-6492899831.1000	145.3185
C108_25t_5w	436	100000	118648952.4799	-8009072.9800	1581.4317
C108_50t_10w	5397	100000	-1001237108.3400	-1125905177.3200	11.0726
C109_100t_20w	7200	19520	11210605305.1199	-6614488230.8200	269.4856
C109_25t_5w	647	100000	36045913.0797	-26031566.9600	238.4700
C109_50t_10w	6451	100000	-1075258981.1794	-1125905761.1800	4.4983
C201_100t_20w	7200	8971	-12596716053.9600	-12596720162.2300	0.0000
C201_25t_5w	3875	100000	-42552163.2401	-45555916.9200	6.5935
C201_50t_10w	7200	34748	-1110321566.2804	-1125905241.3800	1.3841
C202_100t_20w	7200	9052	39322214142.4203	-6055175633.1600	749.3984
C202_25t_5w	7200	26753	-45555938.3608	-45555950.8800	0.0000
C202_50t_10w	7200	9821	-506461044.4007	-1125905361.9600	55.0174
C203_100t_20w	7200	6710	24536869879.3203	-6930623788.8800	454.0355
C203_25t_5w	7200	7578	-45555886.2806	-45555956.2400	0.0001
C203_50t_10w	7200	3816	709052735.5999	-1125905456.0600	162.9762
C204_100t_20w	7202	4592	24609823628.3203	-8657198950.2400	384.2700
C204_25t_5w	7200	5199	-45555876.9005	-45555977.0000	0.0002
C204_50t_10w	7202	3798	93504394.2998	-1125905585.1400	108.3048
C205_100t_20w	7201	5893	9702889586.1598	-12596718687.8400	177.0271
C205_25t_5w	7200	74961	-42552170.0398	-45555926.8000	6.5935
C205_50t_10w	7200	10860	-1125904965.2406	-1125905403.9800	0.0000
C206_100t_20w	7201	3950	31953859516.5598	-7733117210.9600	513.2080
C206_25t_5w	7200	55931	-45555915.5394	-45555926.8000	0.0000
C206_50t_10w	7200	9395	-1118113473.0000	-1125905541.4200	0.6920
C207_100t_20w	7200	3658	24488233802.3396	-7222438653.2400	439.0576
C207_25t_5w	7200	50888	-45555904.1599	-45555933.4200	0.0000
C207_50t_10w	7201	4109	-498668527.7600	-1125905455.9000	55.7095
C208_100t_20w	7201	4384	24536869467.2000	-8657201177.3800	383.4272
C208_25t_5w	7200	35528	-45555924.0793	-45555933.1600	0.0000
C208_50t_10w	7200	7344	-1110321499.1405	-1125905465.6600	1.3841
hh_00_P0	7000	100000	8908128528.0586	6663651008.2900	33.6823
ll1_00_P0	1239	100000	342636882.4399	1338732826.5000	290.7147 *
ll1_01_P0	1257	100000	342635158.9497	1338732826.5000	290.7167 *
ll1_02_P0	1355	100000	373768965.3901	1338732826.5000	258.1712 *
ll1_03_P0	1304	100000	311529698.9099	1338732826.5000	329.7287 *
ll1_04_P0	1284	100000	311531875.5497	1338732826.5000	329.7257 *
ll1_05_P0	1332	100000	435993234.8100	1307660206.1200	199.9267 *
ll1_06_P0	1143	100000	2132305698.5799	1214281948.4400	75.6021
ll1_07_P0	1288	100000	342589125.2799	1338732826.5000	290.7692 *
ll2_00_P0	2046	100000	-179547639.3601	85352916.2700	147.5377 *
ll3_00_P0	1889	100000	-179674136.5400	-198615845.6200	9.5368

Table E.3 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
R101_100t_20w	520	100000	20175837438.8199	6319981502.8600	219.2388
R101_25t_5w	35	100000	60631893.5799	47893886.5800	26.5963
R101_50t_10w	156	100000	1304254280.7799	709051142.3992	83.9436
R102_100t_20w	660	100000	16882100082.3999	14101742003.1800	19.7164
R102_25t_5w	54	100000	38715883.0199	34655177.0000	11.7174
R102_50t_10w	146	100000	1094743006.6199	641955625.9388	70.5325
R103_100t_20w	926	100000	12826397947.8799	12469734030.7400	2.8602
R103_25t_5w	55	100000	34098934.2200	30260803.3400	12.6835
R103_50t_10w	267	100000	959253314.5199	570531480.7600	68.1332
R104_100t_20w	1284	100000	10345962180.4799	14023383931.9600	35.5445 *
R104_25t_5w	75	100000	25977731.5400	29982663.2800	15.4167 *
R104_50t_10w	387	100000	435474486.2800	293491458.8800	48.3772
R105_100t_20w	958	100000	13623487786.5999	8424839701.0800	61.7061
R105_25t_5w	60	100000	47393260.0000	35155829.6400	34.8091
R105_50t_10w	212	100000	967044808.8399	388290958.0600	149.0515
R106_100t_20w	1009	100000	13596468050.6999	8443754034.7200	61.0239
R106_25t_5w	41	100000	34377077.2599	30483330.4600	12.7733
R106_50t_10w	202	100000	897784703.4599	303880567.4400	195.4399
R107_100t_20w	1251	100000	10389194227.7599	11634816307.3400	11.9895 *
R107_25t_5w	72	100000	30038339.4199	22028382.6800	36.3619
R107_50t_10w	272	100000	701259648.6399	372274723.4800	88.3715
R108_100t_20w	1454	100000	5514786001.4799	12369760088.0800	124.3017 *
R108_25t_5w	72	100000	25810858.2400	26033406.5200	0.8622 *
R108_50t_10w	381	100000	377902193.1399	293491505.9600	28.7608
R109_100t_20w	1311	100000	10329750191.7199	6636116061.1200	55.6595
R109_25t_5w	45	100000	42609637.8599	26255954.0600	62.2856
R109_50t_10w	249	100000	764026736.5199	631133985.4200	21.0561
R110_100t_20w	911	100000	11932036220.9798	7622345982.8200	56.5402
R110_25t_5w	50	100000	39272044.5799	34488424.6600	13.8702
R110_50t_10w	198	100000	645418716.5599	636761297.5800	1.3596
R111_100t_20w	858	100000	9481322286.4599	8473476077.2600	11.8941
R111_25t_5w	72	100000	34432709.1399	26144634.7600	31.7008
R111_50t_10w	191	100000	503868711.3799	496942686.1600	1.3937
R112_100t_20w	1074	100000	11172774212.0399	10851236130.0600	2.9631
R112_25t_5w	48	100000	34210250.9800	26311467.9000	30.0203
R112_50t_10w	182	100000	567068465.1599	427249792.1800	32.7252
R201_100t_20w	7200	24726	-529585194.2800	-1383419122.5000	61.7191
R201_25t_5w	1260	100000	-5171474.3401	-5171519.4600	0.0008
R201_50t_10w	7200	34668	-125096977.1800	-125097475.6400	0.0003
R202_100t_20w	7200	28010	2761450612.5399	-778169509.0800	454.8649
R202_25t_5w	1647	100000	-721622.4400	-5171550.1800	86.0463
R202_50t_10w	7200	84451	-56703314.8207	-125097689.9000	54.6727
R203_100t_20w	7200	22682	1148356134.6599	-891653653.5400	228.7894
R203_25t_5w	3627	100000	-721779.9401	-5171614.5200	86.0434
R203_50t_10w	7200	37062	77055320.9399	-125097805.5400	161.5960
R204_100t_20w	7200	16616	1958956445.7196	-1007840112.0800	294.3717
R204_25t_5w	5213	100000	-5060239.9801	-5060450.1000	0.0041
R204_50t_10w	7200	14050	12124162.7199	-125098025.1800	109.6917
R205_100t_20w	7200	17254	1094316349.6598	-910567724.8000	220.1795
R205_25t_5w	2215	100000	-5171703.6203	-5171759.2000	0.0010
R205_50t_10w	7200	27747	-122932840.0400	-125097876.3800	1.7306
R206_100t_20w	7200	16796	2766854778.1198	-964607487.1800	386.8373
R206_25t_5w	4362	100000	-5171681.3800	-5171813.9000	0.0025
R206_50t_10w	7200	37258	-125097654.3802	-125097837.4400	0.0001
R207_100t_20w	7200	20361	2823596034.1598	-1018647805.1000	377.1906
R207_25t_5w	3962	100000	-5171757.0801	-5171794.9600	0.0007
R207_50t_10w	7200	11296	76189580.1999	-125097967.7600	160.9039
R208_100t_20w	7200	13732	335054383.2797	-1042965700.1000	132.1251
R208_25t_5w	4446	100000	-5060452.4001	-5060560.5800	0.0021
R208_50t_10w	7200	10517	-53673163.7199	-125098091.9200	57.0951
R209_100t_20w	7200	11824	1964360621.0598	-1013243421.1000	293.8685
R209_25t_5w	3328	100000	-4559686.1000	-5060483.0600	9.8962
R209_50t_10w	7200	23944	-53240014.2000	-125097999.8800	57.4413
R210_100t_20w	7200	18754	2780364910.6197	-978117773.2200	384.2566
R210_25t_5w	4344	100000	-5171498.6599	-5171642.3200	0.0027
R210_50t_10w	7200	23653	77055070.8399	-125097934.6800	161.5957
R211_100t_20w	7200	11765	6903616225.2398	-942991911.9800	832.0970
R211_25t_5w	3411	100000	-5060349.6600	-5060502.8400	0.0030
R211_50t_10w	7200	27343	-123365886.8602	-122500836.0400	0.7012 *
RC101_100t_20w	569	100000	16965862146.9599	15814809991.8800	7.2783
RC101_25t_5w	67	100000	39438835.2199	27034405.3800	45.8838
RC101_50t_10w	179	100000	848870105.0599	190034189.6200	346.6933
RC102_100t_20w	513	100000	15250092604.6798	15822916717.4800	3.7562 *
RC102_25t_5w	90	100000	26700756.9400	26645148.4800	0.2087
RC102_50t_10w	194	100000	651046170.1400	578756222.9600	12.4905
RC103_100t_20w	802	100000	13653210510.1799	17527878844.9600	28.3791 *
RC103_25t_5w	109	100000	30817186.0398	21805915.8800	41.3248
RC103_50t_10w	230	100000	646284825.1000	767922882.8400	18.8211 *
RC104_100t_20w	925	100000	8054666407.9600	18130424572.1200	125.0921 *
RC104_25t_5w	80	100000	26088953.8400	25810879.2400	1.0773
RC104_50t_10w	242	100000	565770233.9199	498674346.2200	13.4548
RC105_100t_20w	555	100000	14477320336.1998	10907978098.4200	32.7223
RC105_25t_5w	88	100000	35378238.0600	30761388.4200	15.0085
RC105_50t_10w	281	100000	585249275.6200	448893536.2400	30.3759
RC106_100t_20w	874	100000	10440532478.7598	11529438265.0600	10.4296 *
RC106_25t_5w	86	100000	34988937.6400	22417682.5200	56.0774
RC106_50t_10w	196	100000	977001208.3400	515989390.5800	89.3452
RC107_100t_20w	870	100000	16101222667.4800	16638920438.6600	3.3394 *
RC107_25t_5w	49	100000	31039576.4799	30761589.8800	0.9036
RC107_50t_10w	182	100000	518153614.2799	575726019.2200	11.1110 *
RC108_100t_20w	742	100000	12853418676.6198	14823176358.6800	15.3247 *
RC108_25t_5w	78	100000	35155829.7000	30372205.5600	15.7500
RC108_50t_10w	248	100000	199557652.8200	630268451.2000	215.8327 *
RC201_100t_20w	7200	24266	-1388819673.9802	-1378013035.7800	0.7781 *
RC201_25t_5w	852	100000	-5170746.3399	-5171005.4000	0.0050
RC201_50t_10w	7200	74628	-121200055.3401	-125096456.0400	3.1147
RC202_100t_20w	7200	25791	1980573359.1996	-848419159.5200	333.4427

Table E.3 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
RC202_25t_5w	853	100000	-5171400.4000	-5171480.4200	0.0015
RC202_50t_10w	7079	100000	-123365044.3597	-125096955.0600	1.3844
RC203_100t_20w	7200	22280	1129442947.1196	-915968950.1800	223.3058
RC203_25t_5w	4401	100000	-721539.2000	-5171505.3800	86.0477
RC203_50t_10w	7200	38201	9094096.4985	-125096765.1800	107.2696
RC204_100t_20w	7200	13224	1907619772.8397	-1032155821.9000	284.8189
RC204_25t_5w	4833	100000	-5059786.3000	-5060413.3600	0.0123
RC204_50t_10w	7200	9907	12125048.2999	-125097014.9000	109.6925
RC205_100t_20w	7200	23133	1110529849.8197	-856525543.8000	229.6551
RC205_25t_5w	532	100000	-5004455.4399	-5171462.0600	3.2293
RC205_50t_10w	7200	50087	77921766.4399	-125096766.4200	162.2891
RC206_100t_20w	7200	20197	318844738.3798	-905161565.1000	135.2251
RC206_25t_5w	1686	100000	-5171143.4004	-5171397.9400	0.0049
RC206_50t_10w	7200	52093	-119468502.0000	-125097137.7600	4.4994
RC207_100t_20w	7200	18546	3639601716.1598	-907864881.6600	500.8968
RC207_25t_5w	3832	100000	-4559055.2800	-5059890.2000	9.8981
RC207_50t_10w	7200	41747	-119035962.7801	-122499526.2000	2.8274
RC208_100t_20w	7200	14083	6063295696.4599	-978116228.2000	719.8952
RC208_25t_5w	5104	100000	-5060391.7402	-5060744.0000	0.0069
RC208_50t_10w	7200	38949	-121200454.2600	-125096819.1800	3.1146
test150-0-0-0-0_d0_tw0	3456	5219	73735917710.6998	-28349336446.9000	360.0975
test150-0-0-0-0_d0_tw1	7200	20105	-54491442477.5001	-30832491493.7000	43.4177 *
test150-0-0-0-0_d0_tw2	7200	22035	-23727911994.8995	-28832172176.8000	17.7033
test150-0-0-0-0_d0_tw3	7200	23727	6897666282.2002	-27383664735.0000	125.1889
test150-0-0-0-0_d0_tw4	7200	30067	70631974156.4999	-26142085667.0000	370.1849
test250-0-0-0-0_d0_tw0	-	-	-	23650649218.5000	-
test250-0-0-0-0_d0_tw1	7200	16262	-468958093507.5002	-292254131472.0000	37.6801 *
test250-0-0-0-0_d0_tw2	7200	15284	550721722328.1854	-275698693917.1000	299.7549
test250-0-0-0-0_d0_tw3	7200	14021	-214544933324.4147	-266914175565.0000	19.6202
test250-0-0-0-0_d0_tw4	7200	17766	1571077267390.7969	-231776103927.7000	777.8426
test50-0-0-0-0_d0_tw0	7200	17799	-842598576.7999	-842599324.2000	0.0000
test50-0-0-0-0_d0_tw1	7200	47632	-842598443.2999	-842599506.6000	0.0001
test50-0-0-0-0_d0_tw2	7200	68789	-842598048.7999	-842599560.3000	0.0001
test50-0-0-0-0_d0_tw3	7200	72243	-836356284.5000	-842598590.0000	0.7408
test50-0-0-0-0_d0_tw4	7200	59656	-823874047.9999	-842599753.8000	2.2223

E.4 Tabu Search Results: Config. 4

Table E.4: Tabu Search experiments results with parameter configuration 4

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
10_District0	29	1000	71044087988.6800	71563566909.3600	0.7312 *
10_District1	296	1000	8934442503614.9860	8218857439409.0600	8.7066
10_District2	164	1000	46551982147.0399	27771011394.0200	67.6279
10_District3	93	1000	147980306196.2500	139155792160.4600	6.3414
10_District4	908	1000	26247235828066.9450	21291228411410.1000	23.2772
10_District5	932	1000	42185903540403.9500	37249297856014.8000	13.2528
11_District0	48	1000	3934486085.3199	3046672128.3800	29.1404
11_District1	362	1000	7192527346864.9190	6157994611789.7000	16.7998
11_District2	81	1000	14792532946.3500	9214556192.4000	60.5344
11_District3	65	1000	80429174731.5201	61007292651.7800	31.8353
11_District4	420	398	27316194792545.6400	22460659385239.4000	21.6179
11_District5	848	1000	16896435416813.5490	15643005576898.3000	8.0127
12_District0	32	1000	147652850806.6500	115036288619.9000	28.3532
12_District1	737	1000	43251322721596.4900	37877356021322.0000	14.1878
12_District2	58	1000	204319853826.4198	153054550706.1000	33.4947
12_District3	236	1000	252008395524.3999	239836499336.7000	5.0750
12_District4	1784	1000	67678360528545.1900	59129219661289.8000	14.4584
12_District5	1434	1000	73828178354991.5800	65945373929847.0000	11.9535
13_District0	15	1000	180760555855.7400	154315121626.0000	17.1372
13_District1	383	1000	19390813048506.7730	14674175609787.1000	32.1424
13_District2	201	1000	141580059846.0700	126837068235.0100	11.6235
13_District3	705	1000	465854321971.4305	429665639491.2800	8.4225
13_District4	334	583	34082066160556.5860	30033656135618.7000	13.4795
13_District5	410	1000	53058502486492.2340	42764648834839.3000	24.0709
14_District0	19	1000	37910995189.5100	34977146773.4600	8.3879
14_District1	1583	1000	13922893501224.8240	12434388027267.4000	11.9708
14_District2	39	1000	113199977336.8299	90165619840.7900	25.5467
14_District3	244	1000	286753301597.2999	262406322982.9800	9.2783
14_District4	280	333	42319930717404.3200	31546738861338.4000	34.1499
14_District5	1173	982	49108410966425.7700	44524141233501.8000	10.2961
15_District0	22	1000	60908915131.6200	42188641727.2300	44.3727
15_District1	363	1000	13887567215565.1580	12317798430422.9000	12.7439
15_District2	115	1000	89981800622.1499	67421894826.9300	33.4608
15_District3	314	1000	436449737402.5901	463823619193.1600	6.2719 *
15_District4	362	646	30205466091430.0080	22834329911998.4000	32.2809
15_District5	225	325	33576157463980.4840	28700034523627.8000	16.9899
16_District0	23	1000	130552369225.5199	120807470457.0500	8.0664
16_District1	303	1000	14998181921983.5140	12316160134202.9000	21.7764
16_District2	56	1000	102790210506.8200	94504646913.7700	8.7673
16_District3	293	1000	273024373859.2299	210518442436.7200	29.6914
16_District4	674	1000	34201141618648.3050	28327769996973.5000	20.7336
16_District5	749	1000	51196166785094.2400	48522145053303.8000	5.5109
17_District0	19	1000	73131942514.7400	60633779564.4100	20.6125
17_District1	655	1000	12685763159805.3750	12050832937058.4000	5.2687
17_District2	157	1000	121531342483.6900	111787046906.6000	8.7168
17_District3	46	1000	239153190110.8500	178730940805.7800	33.8062

Table E.4 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
17_District4	1320	1000	21610561059438.7660	19092974442252.1000	13.1859
17_District5	412	1000	28273714761962.7730	23886759047749.6000	18.3656
18_District0	5	1000	3050648105.2099	2341527488.6000	30.2845
18_District1	174	1000	14776289420305.5410	13924688690304.7000	6.1157
18_District2	61	1000	7677635119.2600	4417358464.4100	73.8060
18_District3	41	1000	65519767254.4700	70920528499.0700	8.2429 *
18_District4	559	778	19724330089177.8480	18718722507656.5000	5.3722
18_District5	254	1000	15150150777219.4860	12641617614178.6000	19.8434
19_District0	18	1000	47832208781.3799	55462969817.4300	15.9531 *
19_District1	588	1000	25232408796426.6680	22401449014876.9000	12.6373
19_District2	104	1000	173610929825.2799	153280470262.9000	13.2635
19_District3	201	1000	256129174695.5700	243569037679.2000	5.1567
19_District4	2445	1000	129861724782360.6700	106192108824987.0000	22.2894
19_District5	1114	1000	90610807531969.0000	78807991381031.8000	14.9766
1_District0	26	1000	414511188066.4297	430590936121.1900	3.8792 *
1_District1	921	1000	59038322433232.7340	56270206789859.1000	4.9193
1_District2	54	1000	760870638710.4302	672001346398.5800	13.2245
1_District3	436	1000	1158594948599.5605	1031045184732.7300	12.3709
1_District4	3603	782	88015986409409.8800	73732177837593.4000	19.3725
1_District5	1400	1000	103190883734252.5600	91806570717137.9000	12.4003
20_District0	11	1000	101211792729.5600	98421186256.3800	2.8353
20_District1	485	1000	19229197023711.1000	17222315357479.6000	11.6527
20_District2	192	1000	40073505633.6799	27378465012.0200	46.3687
20_District3	383	1000	286248471817.6000	305213925793.8100	6.6255 *
20_District4	1952	1000	30556157284792.6700	24960379220572.1000	22.4186
20_District5	1406	1000	51078407442368.2900	48965990159239.8000	4.3140
21_District0	77	1000	121862046006.5100	120414278382.3300	1.2023
21_District1	402	1000	18687598423451.9960	16893868904739.4000	10.6176
21_District2	189	1000	89851400124.6900	62666013838.2000	43.3813
21_District3	156	1000	976805733287.0194	892017226174.9800	9.5052
21_District4	501	549	26836990955543.4300	24304758516183.0000	10.4186
21_District5	714	1000	51989982622470.8100	45930297567746.2000	13.1932
22_District0	13	1000	155412812002.1598	138906549645.3000	11.8829
22_District1	201	1000	16637453055468.2540	14071948171020.4000	18.2313
22_District2	52	1000	279141000987.0700	226837888943.8600	23.0574
22_District3	333	1000	339965551062.5403	285892462254.7700	18.9137
22_District4	1129	1000	34742341146126.3100	29074924558505.3000	19.4924
22_District5	712	1000	47039138536698.7800	42772519577575.2000	9.9751
23_District0	20	1000	37920541035.5399	32906581132.3800	15.2369
23_District1	878	1000	19553686306408.2850	17211576588480.4000	13.6077
23_District2	50	1000	222085775287.2799	183871311138.5000	20.7832
23_District3	397	1000	234822540354.5498	250619333021.8100	6.7271 *
23_District4	1089	1000	41506283484357.5700	34555727151866.1000	20.1140
23_District5	939	1000	81512772286637.2000	70608343796642.5000	15.4435
24_District0	29	1000	127794166641.5499	105830236823.5000	20.7539
24_District1	301	1000	14050104442788.0570	11389191027676.9000	23.3634
24_District2	128	1000	25910157362.5599	26055867878.7800	0.5623 *
24_District3	381	1000	153413574546.1999	159315579521.2200	3.8471 *
24_District4	1296	1000	27230600919334.0160	24517137738123.8000	11.0676
24_District5	380	1000	28602790614333.3400	24738655441667.6000	15.6198
25_District0	11	1000	1806481293.3700	1255504790.1000	43.8848
25_District1	442	1000	6890897745235.7560	6071232244353.8300	13.5008
25_District2	19	1000	3667525012.0000	2430507030.7800	50.8954
25_District3	63	1000	56516814183.2299	54825213021.3900	3.0854
25_District4	623	1000	23992661886885.8300	20503174454937.0000	17.0192
25_District5	320	1000	12081058658091.7710	11011719621667.4000	9.7109
26_District0	80	1000	237401249526.4201	225380467772.9800	5.3335
26_District1	649	1000	47976908453790.1400	42180173025536.8000	13.7427
26_District2	34	1000	365003699120.1197	282951685653.1000	28.9985
26_District3	770	1000	1221285639579.8296	1171533328289.0400	4.2467
26_District4	1622	1000	122594853392901.8400	105580668249051.0000	16.1148
26_District5	1708	1000	87385202511041.2800	76363123397244.6000	14.4337
27_District0	30	1000	161671621798.6800	162479691641.3000	0.4998 *
27_District1	315	1000	22239448997537.0200	18425601260045.1000	20.6986
27_District2	146	1000	85658869866.1899	61192189969.0700	39.9833
27_District3	133	1000	134916417164.4498	116712931257.8400	15.5968
27_District4	1077	1000	41280478157408.2400	33778988955119.3000	22.2075
27_District5	789	1000	43792124067853.4400	39568513214098.5000	10.6741
28_District0	11	1000	120908555310.4500	102108349093.8000	18.4120
28_District1	634	1000	15643075617476.7600	15128335139531.3000	3.4024
28_District2	53	1000	80576135805.1599	64692968024.3600	24.5516
28_District3	249	1000	837122182158.8696	805624097687.0900	3.9097
28_District4	1223	960	27022467080015.8400	22982825117023.6000	17.5767
28_District5	632	1000	39675458070899.8400	33791534896023.1000	17.4124
29_District0	72	1000	33326728101.2599	29353316624.4500	13.5364
29_District1	173	1000	8270538003155.1100	7484826255380.2500	10.4973
29_District2	149	1000	83239449171.6500	101233312043.5700	21.6169 *
29_District3	307	1000	289675147038.2499	232601979076.2900	24.5368
29_District4	1078	1000	10202501670409.2170	8309110124637.3200	22.7869
29_District5	1279	1000	38201575278526.1700	35670467010142.6000	7.0958
2_District0	45	1000	160408679773.3000	147005506881.3000	9.1174
2_District1	627	1000	44521771049389.4700	39540340925649.8000	12.5983
2_District2	271	1000	239711691525.7900	217829855689.6200	10.0453
2_District3	94	1000	972346621091.5603	858787147589.8100	13.2232
2_District4	2478	1000	71217031447819.6700	58876326066126.1000	20.9603
2_District5	959	1000	111088965128385.5800	95903342380824.9000	15.8342
30_District0	17	1000	16967649427.1899	14856579555.0000	14.2096
30_District1	113	1000	4981652031760.4080	4288991390631.0100	16.1497
30_District2	63	1000	38131062016.3899	28984654012.0000	31.5560
30_District3	170	1000	136962127044.3600	140260859804.8700	2.4084 *
30_District4	688	1000	6588527875154.5260	6033193692111.0700	9.2046
30_District5	386	1000	6146473066871.9660	5491783393822.4800	11.9212
3_District0	19	1000	107537164060.7300	90701681116.3600	18.5613
3_District1	275	1000	19731991879439.8050	17931112587450.7000	10.0433
3_District2	104	1000	181278112603.1000	184423936337.4900	1.7353 *
3_District3	391	1000	927632490655.9303	837894240881.4400	10.7099
3_District4	2983	1000	81441621777835.7700	67042838268903.4000	21.4769

Table E.4 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
3_District5	1458	1000	68650793672125.3800	60566955554266.6000	13.3469
4_District0	6	1000	22790466855.3999	20379141775.5100	11.8323
4_District1	754	1000	17981144122254.7770	16091560843817.3000	11.7426
4_District2	62	1000	24077179304.1799	19771989271.8700	21.7741
4_District3	63	1000	33504443037.0900	31243041764.1500	7.2380
4_District4	2292	1000	53890299945868.2100	44497725128783.0000	21.1079
4_District5	793	1000	35110075062663.1500	30292083288905.3000	15.9051
5_District0	74	1000	108530210898.0600	105305055950.1000	3.0626
5_District1	431	1000	26647092666473.4770	22578812619408.6000	18.0181
5_District2	72	1000	76497862879.7299	49746631100.9200	53.7749
5_District3	384	1000	368151444763.6298	366270241695.4900	0.5136
5_District4	1381	1000	83798881044311.7300	68243049339872.1000	22.7947
5_District5	1062	1000	138169104767029.4700	121803986176569.0000	13.4356
6_District0	14	1000	212929949916.4900	174024536017.0000	22.3562
6_District1	984	1000	28332726145525.2270	24635464198083.6000	15.0078
6_District2	188	1000	272443769103.5498	263920214770.7000	3.2295
6_District3	338	1000	316238009828.5399	272026860685.1900	16.2524
6_District4	1178	1000	39100356623011.2500	32671893526336.1000	19.6758
6_District5	1065	1000	62494974243751.2600	52577668995018.6000	18.8622
7_District0	61	1000	52689510784.6200	53992400830.1800	2.4727 *
7_District1	1087	1000	26730908898720.9300	25530776489059.8000	4.7007
7_District2	31	1000	108798052163.5799	96812988604.0900	12.3796
7_District3	131	1000	734823778262.7499	627921570619.9200	17.0247
7_District4	840	918	37220758140234.5160	30888008083772.9000	20.5022
7_District5	2332	1000	59599457455793.6500	52214576657422.1000	14.1433
8_District0	10	1000	57495503317.6100	47815962594.9100	20.2433
8_District1	466	1000	15334762566736.7190	13654929607346.6000	12.3020
8_District2	122	1000	76471429901.2400	73150043166.7900	4.5405
8_District3	274	1000	440324029530.6004	367102727288.8900	19.9457
8_District4	994	1000	33729940565814.9060	29816748603227.9000	13.1241
8_District5	546	1000	47035070288952.9300	40450402089088.4000	16.2783
9_District0	10	1000	112183055843.6300	100093204504.3000	12.0785
9_District1	231	1000	13712364099437.0000	12178217016824.2000	12.5974
9_District2	33	1000	119904961580.6499	89342594098.5600	34.2080
9_District3	271	1000	1135109174468.4993	911115793606.4100	24.5845
9_District4	1395	1000	45485008173900.7340	37164202673658.1000	22.3893
9_District5	476	1000	30194321664614.4730	25382936280280.4000	18.9551
C101_100t_20w	328	1000	55299138777.7998	-12304901708.1200	549.4073
C101_25t_5w	11	1000	116646365.9200	9512566.6400	1126.2344
C101_50t_10w	53	1000	810344547.3999	-1079154704.1200	175.0906
C102_100t_20w	692	1000	54934369457.7801	-5155408825.8200	1165.5676
C102_25t_5w	27	1000	76596487.9799	-37546115.7600	304.0064
C102_50t_10w	141	1000	3140078105.8398	-1102530180.6200	384.8065
C103_100t_20w	1581	1000	32780671675.6599	-7222438475.8600	553.8726
C103_25t_5w	66	1000	112140918.1599	-43052838.4600	360.4727
C103_50t_10w	347	1000	1336287910.9199	-1110321792.2400	220.3514
C104_100t_20w	3103	1000	3064075113.7200	-8681519205.2400	135.2942
C104_25t_5w	149	1000	114643945.2999	-33540706.8600	441.8053
C104_50t_10w	845	1000	124670837.7799	-1086946666.3200	111.4698
C105_100t_20w	636	1000	18554641574.9399	-12158993843.3000	252.6001
C105_25t_5w	12	1000	122653858.3399	1002190.7200	12138.5745
C105_50t_10w	125	1000	210380233.5199	-1102529614.6600	119.0815
C106_100t_20w	869	1000	17825101737.4402	-5349954009.7600	433.1823
C106_25t_5w	18	1000	116646382.1599	9512566.6400	1126.2345
C106_50t_10w	73	1000	794761334.7799	-1086946589.7000	173.1187
C107_100t_20w	1182	1000	10335156917.3600	-6979258964.6200	248.0838
C107_25t_5w	16	1000	154193283.1200	-2001420.3200	7804.1929
C107_50t_10w	233	1000	155838169.7399	-1125905161.9000	113.8411
C108_100t_20w	1637	1000	32415901528.9200	-6492899831.1000	599.2515
C108_25t_5w	36	1000	121152014.7800	-8009072.9800	1612.6846
C108_50t_10w	242	1000	-416856000.7400	-1125905177.3200	62.9759
C109_100t_20w	2695	1000	24974593783.5199	-6614488230.8200	477.5740
C109_25t_5w	43	1000	76095775.4600	-26031566.9600	392.3211
C109_50t_10w	387	1000	-412960245.9599	-1125905761.1800	63.3219
C201_100t_20w	3104	1000	-12596716053.9600	-12596720162.2300	0.0000
C201_25t_5w	130	1000	-42552000.2200	-45555916.9200	6.5939
C201_50t_10w	634	1000	-1102529624.8800	-1125905241.3800	2.0761
C202_100t_20w	3605	598	39322214142.4203	-6055175633.1600	749.3984
C202_25t_5w	693	1000	-42552100.4800	-45555950.8800	6.5937
C202_50t_10w	1683	1000	81816611.2799	-1125905361.9600	107.2667
C203_100t_20w	3623	337	24536869879.3203	-6930623788.8800	454.0355
C203_25t_5w	3604	726	-39548128.4000	-45555956.2400	13.1877
C203_50t_10w	2131	556	709052735.5999	-1125905456.0600	162.9762
C204_100t_20w	280	22	565879861697.6803	-8657198950.2400	6636.5237
C204_25t_5w	3603	914	-6506909.0600	-45555977.0000	85.7166
C204_50t_10w	1250	252	93504394.2998	-1125905585.1400	108.3048
C205_100t_20w	3606	543	9702889586.1598	-12596718687.8400	177.0271
C205_25t_5w	363	1000	-45555747.0800	-45555926.8000	0.0003
C205_50t_10w	2231	1000	-1118113020.5800	-1125905403.9800	0.6920
C206_100t_20w	3610	346	31953859516.5598	-7733117210.9600	513.2080
C206_25t_5w	374	1000	-6507056.4000	-45555926.8000	85.7163
C206_50t_10w	3060	1000	81816864.3799	-1125905541.4200	107.2667
C207_100t_20w	3602	244	24488233802.3396	-7222438653.2400	439.0576
C207_25t_5w	443	1000	-5005211.4000	-45555933.4200	89.0130
C207_50t_10w	3603	429	-498668527.7600	-1125905455.9000	55.7095
C208_100t_20w	3606	340	24536869467.2000	-8657201177.3800	383.4272
C208_25t_5w	738	1000	-41050213.4000	-45555933.1600	9.8905
C208_50t_10w	3602	668	109087824.3599	-1125905465.6600	109.6888
hh_00_P0	349	1000	147997620950.0366	6663651008.2900	2120.9689
ll1_00_P0	73	1000	8730580967.6700	1338732826.5000	552.1526
ll1_01_P0	77	1000	7096544113.6699	1338732826.5000	430.0941
ll1_02_P0	31	1000	3719583584.4900	1338732826.5000	177.8436
ll1_03_P0	43	1000	3719575520.2600	1338732826.5000	177.8430
ll1_04_P0	60	1000	5353557734.8300	1338732826.5000	299.8973
ll1_05_P0	45	1000	5509237313.5999	1307660206.1200	321.3049
ll1_06_P0	53	1000	8730559873.5401	1214281948.4400	618.9895
ll1_07_P0	73	1000	2132222396.3500	1338732826.5000	59.2716

Table E.4 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
ll2_00_P0	57	1000	-160629321.7401	85352916.2700	153.1365 *
ll3_00_P0	55	1000	-179672936.1900	-198615845.6200	9.5374
R101_100t_20w	12	1000	24272069578.9999	6319981502.8600	284.0528
R101_25t_5w	2	1000	60631889.1799	47893886.5800	26.5963
R101_50t_10w	4	1000	1563979348.3599	709051142.3992	120.5735
R102_100t_20w	28	1000	20921590111.9998	14101742003.1800	48.3617
R102_25t_5w	3	1000	38715883.0199	34655177.0000	11.7174
R102_50t_10w	6	1000	1164868651.7799	641955625.9388	81.4562
R103_100t_20w	55	1000	16830762246.4398	12469734030.7400	34.9729
R103_25t_5w	5	1000	38382009.5399	30260803.3400	26.8373
R103_50t_10w	12	1000	638492764.4999	570531480.7600	11.9119
R104_100t_20w	108	1000	12761550406.4199	14023383931.9600	9.8877 *
R104_25t_5w	5	1000	30316452.5799	29982663.2800	1.1132
R104_50t_10w	27	1000	636328570.8599	293491458.8800	116.8133
R105_100t_20w	27	1000	16898311790.9799	8424839701.0800	100.5772
R105_25t_5w	3	1000	51231298.3800	35155829.6400	45.7263
R105_50t_10w	6	1000	904711001.2199	388290958.0600	132.9982
R106_100t_20w	41	1000	16055287875.3598	8443754034.7200	90.1439
R106_25t_5w	3	1000	34710852.0400	30483330.4600	13.8683
R106_50t_10w	7	1000	832853747.3599	303880567.4400	174.0727
R107_100t_20w	60	1000	14371942267.6998	11634816307.3400	23.5253
R107_25t_5w	4	1000	34098905.0800	22028382.6800	54.7953
R107_50t_10w	12	1000	702125535.5399	372274723.4800	88.6041
R108_100t_20w	83	1000	11923930351.3399	12369760088.0800	3.7389 *
R108_25t_5w	5	1000	30427703.1199	26033406.5200	16.8794
R108_50t_10w	21	1000	638493036.1999	293491505.9600	117.5507
R109_100t_20w	68	1000	16925332156.0598	6636116061.1200	155.0487
R109_25t_5w	3	1000	39605861.5999	26255954.0600	50.8452
R109_50t_10w	9	1000	772684025.1199	631133985.4200	22.4278
R110_100t_20w	53	1000	15209562146.2398	7622345982.8200	99.5391
R110_25t_5w	3	1000	39550221.2199	34488424.6600	14.6767
R110_50t_10w	8	1000	699528307.6399	636761297.5800	9.8572
R111_100t_20w	52	1000	16014758164.0998	8473476077.2600	88.9986
R111_25t_5w	4	1000	34265854.8400	26144634.7600	31.0626
R111_50t_10w	12	1000	571397107.7400	496942686.1600	14.9824
R112_100t_20w	79	1000	12780464289.4199	10851236130.0600	17.7788
R112_25t_5w	3	1000	34265807.6199	26311467.9000	30.2314
R112_50t_10w	9	1000	564038204.4999	427249792.1800	32.0160
R201_100t_20w	813	1000	-570115623.6601	-1383419122.5000	58.7893
R201_25t_5w	42	1000	-5004448.0600	-5171519.4600	3.2306
R201_50t_10w	435	1000	-124231486.3201	-125097475.6400	0.6922
R202_100t_20w	967	1000	3545030516.1598	-778169509.0800	555.5601
R202_25t_5w	51	1000	-554585.4200	-5171550.1800	89.2762
R202_50t_10w	284	1000	79652564.4599	-125097689.9000	163.6722
R203_100t_20w	1279	1000	1907618732.0198	-891653653.5400	313.9416
R203_25t_5w	177	1000	-387838.3000	-5171614.5200	92.5006
R203_50t_10w	720	1000	8228328.8999	-125097805.5400	106.5775
R204_100t_20w	2564	1000	1097018547.0399	-1007840112.0800	208.8484
R204_25t_5w	178	1000	-721375.6599	-5060450.1000	85.7448
R204_50t_10w	1457	1000	-56270063.8600	-125098025.1800	55.0192
R205_100t_20w	1440	1000	1072700765.2797	-910567724.8000	217.8057
R205_25t_5w	94	1000	-4726354.4799	-5171759.2000	8.6122
R205_50t_10w	842	1000	-58001524.5400	-125097876.3800	53.6350
R206_100t_20w	1569	1000	3545030477.8998	-964607487.1800	467.5101
R206_25t_5w	157	1000	-4893151.3600	-5171813.9000	5.3881
R206_50t_10w	684	1000	77055144.5399	-125097837.4400	161.5959
R207_100t_20w	1737	1000	2726324300.8598	-1018647805.1000	367.6415
R207_25t_5w	173	1000	-721463.3399	-5171794.9600	86.0500
R207_50t_10w	899	1000	9959520.2199	-125097967.7600	107.9613
R208_100t_20w	2412	1000	1078104353.7397	-1042965700.1000	203.3691
R208_25t_5w	169	1000	-554730.5200	-5060560.5800	89.0381
R208_50t_10w	1556	1000	-54106033.9599	-125098091.9200	56.7491
R209_100t_20w	2036	1000	1896810658.4798	-1013243421.1000	287.2018
R209_25t_5w	106	1000	-498920.1600	-5060483.0600	90.1408
R209_50t_10w	792	1000	-53672874.0800	-125097999.8800	57.0953
R210_100t_20w	1937	1000	4374544627.5798	-978117773.2200	547.2410
R210_25t_5w	141	1000	-554618.0000	-5171642.3200	89.2757
R210_50t_10w	858	1000	77055144.8199	-125097934.6800	161.5958
R211_100t_20w	2775	1000	6852278368.3198	-942991911.9800	826.6529
R211_25t_5w	81	1000	-610148.7000	-5060502.8400	87.9429
R211_50t_10w	714	1000	15154369.8599	-122500836.0400	112.3708
RC101_100t_20w	20	1000	19440894051.4398	15814809991.8800	22.9284
RC101_25t_5w	3	1000	39438826.4000	27034405.3800	45.8838
RC101_50t_10w	5	1000	853631480.2999	190034189.6200	349.1988
RC102_100t_20w	26	1000	16903716519.2398	15822916717.4800	6.8305
RC102_25t_5w	4	1000	34766435.1399	26645148.4800	30.4794
RC102_50t_10w	6	1000	978300098.2199	578756222.9600	69.0349
RC103_100t_20w	45	1000	17722422857.8798	17527878844.9600	1.1099
RC103_25t_5w	4	1000	34432795.2599	21805915.8800	57.9057
RC103_50t_10w	8	1000	832854072.8999	767922882.8400	8.4554
RC104_100t_20w	66	1000	15982334747.8598	18130424572.1200	13.4404 *
RC104_25t_5w	5	1000	30372053.2599	25810879.2400	17.6715
RC104_50t_10w	12	1000	833287060.9200	498674346.2200	67.1004
RC105_100t_20w	23	1000	23499298622.4998	10907978098.4200	115.4322
RC105_25t_5w	4	1000	39160876.6399	30761388.4200	27.3052
RC105_50t_10w	6	1000	794327894.1199	448893536.2400	76.9524
RC106_100t_20w	35	1000	17749442426.2399	11529438265.0600	53.9488
RC106_25t_5w	3	1000	34877723.9400	22417682.5200	55.5813
RC106_50t_10w	6	1000	776580052.8199	515989390.5800	50.5031
RC107_100t_20w	52	1000	20997246672.5398	16638920438.6600	26.1935
RC107_25t_5w	3	1000	39049565.4799	30761589.8800	26.9426
RC107_50t_10w	6	1000	705588703.7799	575726019.2200	22.5563
RC108_100t_20w	62	1000	13591064183.8599	14823176358.6800	9.0656 *
RC108_25t_5w	3	1000	34877710.2599	30372205.5600	14.8343
RC108_50t_10w	9	1000	508197612.6599	630268451.2000	24.0203 *
RC201_100t_20w	891	1000	-1394223574.2202	-1378013035.7800	1.1626 *
RC201_25t_5w	34	1000	-5003794.9400	-5171005.4000	3.2336

Table E.4 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
RC201_50t_10w	255	1000	-123364159.9800	-125096456.0400	1.3847
RC202_100t_20w	710	1000	2734431573.8397	-848419159.5200	422.2972
RC202_25t_5w	57	1000	-609805.6999	-5171480.4200	88.2082
RC202_50t_10w	249	1000	77921659.6798	-125096955.0600	162.2890
RC203_100t_20w	1440	1000	1083509143.1997	-915968950.1800	218.2910
RC203_25t_5w	142	1000	-554284.4599	-5171505.3800	89.2819
RC203_50t_10w	510	1000	77055884.3598	-125096765.1800	161.5970
RC204_100t_20w	2385	1000	1088914102.8798	-1032155821.9000	205.4990
RC204_25t_5w	145	1000	-275958.5399	-5060413.3600	94.5467
RC204_50t_10w	864	1000	12990559.6799	-125097014.9000	110.3843
RC205_100t_20w	857	1000	1902216215.3399	-856525543.8000	322.0851
RC205_25t_5w	45	1000	-387368.5399	-5171462.0600	92.5094
RC205_50t_10w	304	1000	12125163.7398	-125096766.4200	109.6926
RC206_100t_20w	1229	1000	-559305450.6601	-905161565.1000	38.2093
RC206_25t_5w	67	1000	-609644.1199	-5171397.9400	88.2112
RC206_50t_10w	426	1000	-122065416.8601	-125097137.7600	2.4234
RC207_100t_20w	1777	1000	1931938179.4799	-907864881.6600	312.8001
RC207_25t_5w	80	1000	168827.2399	-5059890.2000	103.3365
RC207_50t_10w	552	1000	-119468890.7601	-122499526.2000	2.4739
RC208_100t_20w	2392	1000	4393459958.3199	-978116228.2000	549.1756
RC208_25t_5w	105	1000	-4781866.9399	-5060744.0000	5.5105
RC208_50t_10w	603	1000	-51939735.4400	-125096819.1800	58.4803
test150-0-0-0-0_d0_tw0	3602	684	101050622678.1997	-28349336446.9000	456.4479
test150-0-0-0-0_d0_tw1	1739	1000	-24141770230.7996	-30832491493.7000	21.7002
test150-0-0-0-0_d0_tw2	1237	1000	7242547420.0989	-28832172176.8000	125.1196
test150-0-0-0-0_d0_tw3	930	1000	6897666282.2002	-27383664735.0000	125.1889
test150-0-0-0-0_d0_tw4	1060	1000	132434939872.1001	-26142085667.0000	606.5966
test250-0-0-0-0_d0_tw0	-	-	-	23650649218.5000	-
test250-0-0-0-0_d0_tw1	-	-	-	-292254131472.0000	-
test250-0-0-0-0_d0_tw2	-	-	-	-275698693917.1000	-
test250-0-0-0-0_d0_tw3	-	-	-	-266914175565.0000	-
test250-0-0-0-0_d0_tw4	-	-	-	-231776103927.7000	-
test50-0-0-0-0_d0_tw0	1062	969	-842598897.8999	-842599324.2000	0.0000
test50-0-0-0-0_d0_tw1	322	1000	-842598628.3999	-842599506.6000	0.0001
test50-0-0-0-0_d0_tw2	300	1000	-842598505.9000	-842599560.3000	0.0001
test50-0-0-0-0_d0_tw3	235	1000	-842597557.8000	-842598590.0000	0.0001
test50-0-0-0-0_d0_tw4	256	1000	-842598879.0000	-842599753.8000	0.0001

E.5 Tabu Search Results: Config. 5

Table E.5: Tabu Search experiments results with parameter configuration 5

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
10_District0	51	1000	77610314919.4900	71563566909.3600	8.4494
10_District1	375	1000	9859295456013.5200	8218857439409.0600	19.9594
10_District2	648	1000	52718867462.9899	27771011394.0200	89.8341
10_District3	136	1000	135190120879.8601	139155792160.4600	2.9334 *
10_District4	976	1000	28824022749673.9530	21291228411410.1000	35.3798
10_District5	828	1000	42657438984971.2600	37249297856014.8000	14.5187
11_District0	85	1000	4231488549.9500	3046672128.3800	38.8888
11_District1	931	1000	7519784104587.9960	6157994611789.7000	22.1141
11_District2	87	1000	15673266449.6800	9214556192.4000	70.0924
11_District3	198	1000	72572873852.9499	61007292651.7800	18.9577
11_District4	1673	1000	28327341544190.6520	22460659385239.4000	26.1198
11_District5	1169	1000	17354107747842.5300	15643005576898.3000	10.9384
12_District0	70	1000	131534476913.1199	115036288619.9000	14.3417
12_District1	1303	1000	46524033514596.0800	37877356021322.0000	22.8280
12_District2	251	1000	189900328960.9298	153054550706.1000	24.0736
12_District3	467	1000	281321238143.1900	239836499336.7000	17.2970
12_District4	1610	1000	69976301190337.2660	59129219661289.8000	18.3447
12_District5	1537	1000	76856001634357.9400	65945373929847.0000	16.5449
13_District0	34	1000	176227768554.3200	154315121626.0000	14.1999
13_District1	631	1000	19871772650002.9260	14674175609787.1000	35.4200
13_District2	378	1000	154666155183.3599	126837068235.0100	21.9408
13_District3	1234	1000	513972607706.4299	429665639491.2800	19.6215
13_District4	1213	1000	34944914457256.2730	30033656135618.7000	16.3525
13_District5	862	1000	56929657841132.4100	42764648834839.3000	33.1231
14_District0	72	1000	36580402569.8800	34977146773.4600	4.5837
14_District1	1067	1000	14524429777197.5840	12434388027267.4000	16.8085
14_District2	152	1000	117890501676.8599	90165619840.7900	30.7488
14_District3	419	1000	286452721000.2198	262406322982.9800	9.1638
14_District4	1025	1000	43274108721617.1600	31546738861338.4000	37.1745
14_District5	906	1000	50752031506542.5160	44524141233501.8000	13.9876
15_District0	67	1000	58854582696.0899	42188641727.2300	39.5033
15_District1	1102	1000	14069935349289.4470	12317798430422.9000	14.2244
15_District2	545	1000	94700044092.7600	67421894826.9300	40.4588
15_District3	901	1000	452400421567.4308	463823619193.1600	2.5250 *
15_District4	875	1000	32194102455076.9380	22834329911998.4000	40.9899
15_District5	878	1000	33961376146576.4570	28700034523627.8000	18.3321
16_District0	26	1000	126976451412.8699	120807470457.0500	5.1064
16_District1	502	1000	15793501181685.3070	12316160134202.9000	28.2339
16_District2	137	1000	119935584173.3399	94504646913.7700	26.9097
16_District3	315	1000	273024373859.2299	210518442436.7200	29.6914
16_District4	1172	1000	35952100828187.3600	28327769996973.5000	26.9146
16_District5	1715	1000	54102712144284.7300	48522145053303.8000	11.5010
17_District0	29	1000	68887283461.8000	60633779564.4100	13.6120
17_District1	659	1000	13404875141608.5880	12050832937058.4000	11.2360

Table E.5 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
17_District2	337	1000	137388519267.1899	111787046906.6000	22.9020
17_District3	106	1000	230540177487.3401	178730940805.7800	28.9872
17_District4	1218	1000	22294458031455.2340	19092974442252.1000	16.7678
17_District5	911	1000	28598180584785.0620	23886759047749.6000	19.7239
18_District0	6	1000	2903112973.2800	2341527488.6000	23.9837
18_District1	362	1000	15347660439622.5760	13924688690304.7000	10.2190
18_District2	145	1000	6910511088.1000	4417358464.4100	56.4398
18_District3	60	1000	72389156352.7300	70920528499.0700	2.0708
18_District4	883	1000	21449713637219.9300	18718722507656.5000	14.5896
18_District5	419	1000	15543852583612.3090	12641617614178.6000	22.9577
19_District0	51	1000	56479422814.1199	55462969817.4300	1.8326
19_District1	1003	1000	26413497392708.1800	22401449014876.9000	17.9097
19_District2	301	1000	180362880232.3901	153280470262.9000	17.6685
19_District3	380	1000	274547752181.1901	243569037679.2000	12.7186
19_District4	3601	928	129861724782360.6700	106192108824987.0000	22.2894
19_District5	1048	1000	92374621349656.3100	78807991381031.8000	17.2147
1_District0	86	1000	435637504247.2400	430590936121.1900	1.1720
1_District1	1401	1000	64113401266731.5800	56270206789859.1000	13.9384
1_District2	244	1000	759126474932.5096	672001346398.5800	12.9650
1_District3	620	1000	1245277165285.8801	1031045184732.7300	20.7781
1_District4	3113	1000	89000469461106.8300	73732177837593.4000	20.7077
1_District5	1262	1000	104180529283572.0200	91806570717137.9000	13.4782
20_District0	25	1000	101142027531.8000	98421186256.3800	2.7644
20_District1	633	1000	20166473748877.4140	17222315357479.6000	17.0950
20_District2	526	1000	39835580284.6499	27378465012.0200	45.4996
20_District3	806	1000	332285164824.6803	305213925793.8100	8.8695
20_District4	1346	1000	30556157284792.6700	24960379220572.1000	22.4186
20_District5	1161	1000	54096789514340.4500	48965990159239.8000	10.4782
21_District0	113	1000	130523686478.3801	120414278382.3300	8.3955
21_District1	953	1000	20356277402674.2970	16893868904739.4000	20.4950
21_District2	323	1000	103704174438.7400	62666013838.2000	65.4871
21_District3	373	1000	1083818307996.6499	892017226174.9800	21.5019
21_District4	1424	1000	27599158275505.6500	24304758516183.0000	13.5545
21_District5	964	1000	57789774017343.4800	45930297567746.2000	25.8205
22_District0	19	1000	155463600461.7400	138906549645.3000	11.9195
22_District1	420	1000	17528245590773.4550	14071948171020.4000	24.5616
22_District2	97	1000	288789611709.5702	226837888943.8600	27.3110
22_District3	569	1000	389586838914.7998	285892462254.7700	36.2704
22_District4	988	1000	36430067637167.2660	29074924558505.3000	25.2972
22_District5	833	1000	49609639676068.9840	42772519577575.2000	15.9848
23_District0	22	1000	35374081297.1500	32906581132.3800	7.4985
23_District1	1199	1000	20056201128206.6600	17211576588480.4000	16.5273
23_District2	141	1000	237060608258.7398	183871311138.5000	28.9274
23_District3	1444	1000	286075456109.0404	250619333021.8100	14.1474
23_District4	1447	1000	44124735609381.3750	34555727151866.1000	27.6915
23_District5	968	1000	82318311632011.2200	70608343796642.5000	16.5843
24_District0	68	1000	120568897844.4298	105830236823.5000	13.9267
24_District1	814	1000	14601551656495.2070	11389191027676.9000	28.2053
24_District2	270	1000	29036331044.5499	26055867878.7800	11.4387
24_District3	580	1000	176442969278.3299	159315579521.2200	10.7506
24_District4	1561	1000	28069693758227.9500	24517137738123.8000	14.4900
24_District5	487	1000	29285060431772.2460	24738655441667.6000	18.3777
25_District0	35	1000	1616801116.4000	1255504790.1000	28.7769
25_District1	643	1000	7110320772712.6380	6071232244353.8300	17.1149
25_District2	23	1000	3087105815.5599	2430507030.7800	27.0148
25_District3	124	1000	62741036683.0999	54825213021.3900	14.4382
25_District4	1193	1000	25996861803541.5400	20503174454937.0000	26.7943
25_District5	747	1000	12241327060108.4200	11011719621667.4000	11.1663
26_District0	115	1000	237401249998.5500	225380467772.9800	5.3335
26_District1	649	1000	48987132161122.1500	42180173025536.8000	16.1378
26_District2	114	1000	364719781615.1997	282951685653.1000	28.8982
26_District3	735	1000	1292287659214.3389	1171533328289.0400	10.3073
26_District4	2248	1000	126120081289807.2700	105580668249051.0000	19.4537
26_District5	1606	1000	88272702517243.4700	76363123397244.6000	15.5959
27_District0	61	1000	167039512085.8001	162479691641.3000	2.8063
27_District1	662	1000	24319276435195.9730	18425601260045.1000	31.9863
27_District2	583	1000	85658870203.8001	61192189969.0700	39.9833
27_District3	225	1000	134886328661.2399	116712931257.8400	15.5710
27_District4	1279	1000	42830700078064.5100	33778988955119.3000	26.7968
27_District5	1138	1000	46799178020401.0300	39568513214098.5000	18.2737
28_District0	24	1000	124370598128.5300	102108349093.8000	21.8025
28_District1	1359	1000	16751294190753.8380	15128335139531.3000	10.7279
28_District2	174	1000	80112017557.5800	64692968024.3600	23.8341
28_District3	527	1000	957802714175.6798	805624097687.0900	18.8895
28_District4	1188	1000	28571951497097.7540	22982825117023.6000	24.3187
28_District5	779	1000	41410160521065.4840	33791534896023.1000	22.5459
29_District0	84	1000	36311987243.3600	29353316624.4500	23.7065
29_District1	349	1000	8894568991971.7270	7484826255380.2500	18.8346
29_District2	340	1000	103115397025.0699	101233312043.5700	1.8591
29_District3	590	1000	289675147038.2499	232601979076.2900	24.5368
29_District4	1202	1000	10869563929843.5680	8309110124637.3200	30.8150
29_District5	1423	1000	40098034357696.8600	35670467010142.6000	12.4124
2_District0	115	1000	172278308041.7100	147005506881.3000	17.1917
2_District1	765	1000	45459857429067.4450	39540340925649.8000	14.9708
2_District2	289	1000	213471461825.7700	217829855689.6200	2.0416 *
2_District3	256	1000	1104883688153.2100	858787147589.8100	28.6562
2_District4	2097	1000	74239030120333.2500	58876326066126.1000	26.0931
2_District5	1212	1000	112136908652761.6000	95903342380824.9000	16.9270
30_District0	18	1000	16304517979.9500	14856579555.0000	9.7461
30_District1	465	1000	4897044477337.7070	4288991390631.0100	14.1770
30_District2	231	1000	42341563154.6299	28984654012.0000	46.0826
30_District3	179	1000	130882107697.1599	140260859804.8700	7.1658 *
30_District4	1089	1000	6957178114976.4290	6033193692111.0700	15.3150
30_District5	1006	1000	6652589252190.8170	5491783393822.4800	21.1371
3_District0	22	1000	107681232956.4300	90701681116.3600	18.7202
3_District1	714	1000	21669256497445.5620	17931112587450.7000	20.8472
3_District2	376	1000	196614004269.0600	184423936337.4900	6.6098

Table E.5 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
3_District3	617	1000	1028904180650.5793	837894240881.4400	22.7964
3_District4	2994	1000	84273459822075.7700	67042838268903.4000	25.7009
3_District5	1261	1000	71450369212445.9400	60566955554266.6000	17.9692
4_District0	18	1000	21123207758.1200	20379141775.5100	3.6511
4_District1	803	1000	18201531064023.1170	16091560843817.3000	13.1122
4_District2	135	1000	23954524801.7300	19771989271.8700	21.1538
4_District3	154	1000	37506527480.3199	31243041764.1500	20.0476
4_District4	2250	1000	55827883212635.5600	44497725128783.0000	25.4623
4_District5	1014	1000	35485732192312.5100	30292083288905.3000	17.1452
5_District0	88	1000	111830368871.1999	105305055950.1000	6.1965
5_District1	549	1000	26958729717452.4450	22578812619408.6000	19.3983
5_District2	247	1000	80738320475.2001	49746631100.9200	62.2990
5_District3	743	1000	368339564893.9703	366270241695.4900	0.5649
5_District4	1547	1000	87052360312381.4500	68243049339872.1000	27.5622
5_District5	1266	1000	144263825380312.4000	121803986176569.0000	18.4393
6_District0	27	1000	213300477607.6500	174024536017.0000	22.5691
6_District1	1176	1000	29757855339023.4100	24635464198083.6000	20.7927
6_District2	421	1000	314596614955.7901	263920214770.7000	19.2014
6_District3	418	1000	306714848859.6501	272026860685.1900	12.7516
6_District4	1128	1000	41017633874659.3900	32671893526336.1000	25.5440
6_District5	1159	1000	65047160187885.1250	52577668995018.6000	23.7163
7_District0	77	1000	59560706885.9500	53992400830.1800	10.3131
7_District1	1575	1000	28469891691201.0230	25530776489059.8000	11.5120
7_District2	74	1000	113460433555.3699	96812988604.0900	17.1954
7_District3	171	1000	800391610726.7496	627921570619.9200	27.4668
7_District4	1123	1000	38983888110125.7900	30888008083772.9000	26.2104
7_District5	1918	1000	59599457455793.6500	52214576657422.1000	14.1433
8_District0	32	1000	57657098453.0400	47815962594.9100	20.5812
8_District1	898	1000	15922201158605.7680	13654929607346.6000	16.6040
8_District2	190	1000	79434821039.0800	73150043166.7900	8.5916
8_District3	446	1000	426128062720.5102	367102727288.8900	16.0786
8_District4	995	1000	35399491028950.2900	29816748603227.9000	18.7235
8_District5	613	1000	50474401568700.1000	40450402089088.4000	24.7809
9_District0	12	1000	112307266806.7400	100093204504.3000	12.2026
9_District1	439	1000	14420982580619.1700	12178217016824.2000	18.4162
9_District2	97	1000	115774157223.7899	89342594098.5600	29.5845
9_District3	405	1000	1254004008182.2605	911115793606.4100	37.6338
9_District4	1523	1000	48344285222867.3100	37164202673658.1000	30.0829
9_District5	707	1000	32702225945411.4060	25382936280280.4000	28.8354
C101_100t_20w	205	1000	91727502838.5603	-12304901708.1200	845.4549
C101_25t_5w	13	1000	116646360.5600	9512566.6400	1126.2343
C101_50t_10w	46	1000	2025857421.8399	-1079154704.1200	287.7263
C102_100t_20w	862	1000	98901313024.3402	-5155408825.8200	2018.3990
C102_25t_5w	29	1000	116145843.0199	-37546115.7600	409.3418
C102_50t_10w	190	1000	3140078105.8398	-1102530180.6200	384.8065
C103_100t_20w	2413	1000	39930163004.1205	-7222438475.8600	652.8626
C103_25t_5w	171	1000	111139618.8199	-43052838.4600	358.1470
C103_50t_10w	1014	1000	3743938477.6599	-1110321792.2400	437.1940
C104_100t_20w	3600	836	17314423586.3605	-8681519205.2400	299.4400
C104_25t_5w	354	1000	113142010.7999	-33540706.8600	437.3274
C104_50t_10w	2351	1000	1305120952.8799	-1086946666.3200	220.0722
C105_100t_20w	516	1000	77185339028.9999	-12158993843.3000	734.8003
C105_25t_5w	20	1000	154693876.4199	1002190.7200	15335.5726
C105_50t_10w	125	1000	3186828351.9399	-1102529614.6600	389.0469
C106_100t_20w	677	1000	54739825368.8202	-5349954009.7600	1123.1831
C106_25t_5w	22	1000	116646508.9600	9512566.6400	1126.2359
C106_50t_10w	77	1000	1379142584.3799	-1086946589.7000	226.8822
C107_100t_20w	929	1000	69525168735.1598	-6979258964.6200	1096.1683
C107_25t_5w	32	1000	151189564.6599	-2001420.3200	7654.1136
C107_50t_10w	232	1000	3747834654.6599	-1125905161.9000	432.8730
C108_100t_20w	1387	1000	39735619043.3798	-6492899831.1000	711.9857
C108_25t_5w	50	1000	110138370.2199	-8009072.9800	1475.1700
C108_50t_10w	368	1000	2551800875.0799	-1125905177.3200	326.6443
C109_100t_20w	2629	1000	47006701457.3999	-6614488230.8200	810.6627
C109_25t_5w	65	1000	77597819.7000	-26031566.9600	398.0912
C109_50t_10w	694	1000	2524529654.7199	-1125905761.1800	324.2221
C201_100t_20w	1212	1000	-12596716053.9600	-12596720162.2300	0.0000
C201_25t_5w	134	1000	-45555894.0200	-45555916.9200	0.0000
C201_50t_10w	337	1000	-506460875.0600	-1125905241.3800	55.0174
C202_100t_20w	3604	547	39322214142.4203	-6055175633.1600	749.3984
C202_25t_5w	742	1000	-45555860.5800	-45555950.8800	0.0001
C202_50t_10w	2349	1000	685677219.4399	-1125905361.9600	160.9000
C203_100t_20w	3604	214	24536869879.3203	-6930623788.8800	454.0355
C203_25t_5w	3607	595	-42551987.5001	-45555956.2400	6.5940
C203_50t_10w	2727	370	709052735.5999	-1125905456.0600	162.9762
C204_100t_20w	277	22	565879861697.6803	-8657198950.2400	6636.5237
C204_25t_5w	3607	521	-6506972.4800	-45555977.0000	85.7165
C204_50t_10w	844	102	93504394.2998	-1125905585.1400	108.3048
C205_100t_20w	3604	984	9702889586.1598	-12596718687.8400	177.0271
C205_25t_5w	343	1000	-45555905.7600	-45555926.8000	0.0000
C205_50t_10w	1045	1000	-522044220.5000	-1125905403.9800	53.6333
C206_100t_20w	3602	404	31953859516.5598	-7733117210.9600	513.2080
C206_25t_5w	523	1000	-45555838.6800	-45555926.8000	0.0001
C206_50t_10w	2119	1000	677885628.8399	-1125905541.4200	160.2080
C207_100t_20w	3607	272	24488233802.3396	-7222438653.2400	439.0576
C207_25t_5w	565	1000	-44053971.9800	-45555933.4200	3.2969
C207_50t_10w	3604	533	-498668527.7600	-1125905455.9000	55.7095
C208_100t_20w	3617	307	24536869467.2000	-8657201177.3800	383.4272
C208_25t_5w	791	1000	-45555831.4600	-45555933.1600	0.0002
C208_50t_10w	3434	1000	697364998.0999	-1125905465.6600	161.9381
hh_00_P0	328	1000	855509231017.3870	6663651008.2900	12738.4459
ll1_00_P0	68	1000	10395714853.6900	1338732826.5000	676.5339
ll1_01_P0	69	1000	8761695462.9999	1338732826.5000	554.4767
ll1_02_P0	71	1000	8761705984.6401	1338732826.5000	554.4775
ll1_03_P0	71	1000	10364606753.6001	1338732826.5000	674.2102
ll1_04_P0	75	1000	10333442162.1100	1338732826.5000	671.8823
ll1_05_P0	66	1000	15360038245.1100	1307660206.1200	1074.6199

Table E.5 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
ll1_06_P0	71	1000	12029720714.7801	1214281948.4400	890.6859
ll1_07_P0	71	1000	5431391939.7000	1338732826.5000	305.7114
ll2_00_P0	97	1000	402247583.6998	85352916.2700	371.2757
ll3_00_P0	94	1000	-179672936.1900	-198615845.6200	9.5374
R101_100t_20w	18	1000	28343983676.1399	6319981502.8600	348.4820
R101_25t_5w	3	1000	56738175.0199	47893886.5800	18.4664
R101_50t_10w	5	1000	1309881552.8798	709051142.3992	84.7372
R102_100t_20w	48	1000	20921590111.9998	14101742003.1800	48.3617
R102_25t_5w	4	1000	38715883.0199	34655177.0000	11.7174
R102_50t_10w	11	1000	973537982.1599	641955625.9388	51.6519
R103_100t_20w	136	1000	19284178221.0998	12469734030.7400	54.6478
R103_25t_5w	9	1000	33932065.1800	30260803.3400	12.1320
R103_50t_10w	19	1000	896919196.0999	570531480.7600	57.2076
R104_100t_20w	196	1000	18470876623.0598	14023383931.9600	31.7148
R104_25t_5w	8	1000	30372075.3399	29982663.2800	1.2987
R104_50t_10w	42	1000	760563926.4399	293491458.8800	159.1434
R105_100t_20w	50	1000	24201817987.1599	8424839701.0800	187.2674
R105_25t_5w	4	1000	47782625.1399	35155829.6400	35.9166
R105_50t_10w	8	1000	1041932251.5599	388290958.0600	168.3380
R106_100t_20w	86	1000	23345284257.8198	8443754034.7200	176.4799
R106_25t_5w	5	1000	38493331.4999	30483330.4600	26.2766
R106_50t_10w	13	1000	901247796.8799	303880567.4400	196.5796
R107_100t_20w	149	1000	20113692439.8597	11634816307.3400	72.8750
R107_25t_5w	8	1000	34098937.7599	22028382.6800	54.7954
R107_50t_10w	24	1000	832420808.0200	372274723.4800	123.6039
R108_100t_20w	223	1000	16028268646.7598	12369760088.0800	29.5762
R108_25t_5w	9	1000	26033326.4599	26033406.5200	0.0003 *
R108_50t_10w	49	1000	699095504.0799	293491505.9600	138.1995
R109_100t_20w	99	1000	18454664083.6199	6636116061.1200	178.0943
R109_25t_5w	5	1000	38437744.6000	26255954.0600	46.3962
R109_50t_10w	15	1000	838913938.4600	631133985.4200	32.9216
R110_100t_20w	129	1000	20138010281.6998	7622345982.8200	164.1970
R110_25t_5w	5	1000	39550222.5599	34488424.6600	14.6768
R110_50t_10w	14	1000	701259778.5999	636761297.5800	10.1291
R111_100t_20w	129	1000	23372304372.9398	8473476077.2600	175.8290
R111_25t_5w	5	1000	34432714.3800	26144634.7600	31.7008
R111_50t_10w	21	1000	831122241.4399	496942686.1600	67.2471
R112_100t_20w	226	1000	20908080380.0198	10851236130.0600	92.6792
R112_25t_5w	5	1000	34655197.5000	26311467.9000	31.7113
R112_50t_10w	20	1000	638060060.4399	427249792.1800	49.3412
R201_100t_20w	422	1000	253994839.7598	-1383419122.5000	118.3599
R201_25t_5w	52	1000	-5004443.9200	-5171519.4600	3.2306
R201_50t_10w	171	1000	-124231486.3201	-125097475.6400	0.6922
R202_100t_20w	1136	1000	4379948786.1398	-778169509.0800	662.8527
R202_25t_5w	93	1000	-721464.7200	-5171550.1800	86.0493
R202_50t_10w	283	1000	146748317.1199	-125097689.9000	217.3069
R203_100t_20w	1935	1000	1907618989.9997	-891653653.5400	313.9417
R203_25t_5w	267	1000	-554393.6000	-5171614.5200	89.2800
R203_50t_10w	861	1000	77055479.7399	-125097805.5400	161.5961
R204_100t_20w	3601	905	2726324834.8998	-1007840112.0800	370.5116
R204_25t_5w	363	1000	-387745.0599	-5060450.1000	92.3377
R204_50t_10w	2657	1000	78354311.8399	-125098025.1800	162.6343
R205_100t_20w	1322	1000	1896810807.6597	-910567724.8000	308.3107
R205_25t_5w	108	1000	-721679.2999	-5171759.2000	86.0457
R205_50t_10w	519	1000	9094043.7598	-125097876.3800	107.2695
R206_100t_20w	2144	1000	5193250386.1198	-964607487.1800	638.3796
R206_25t_5w	162	1000	-4893465.1600	-5171813.9000	5.3820
R206_50t_10w	617	1000	77921377.8798	-125097837.4400	162.2883
R207_100t_20w	2960	1000	3555838829.4997	-1018647805.1000	449.0744
R207_25t_5w	215	1000	-4893192.6600	-5171794.9600	5.3869
R207_50t_10w	1435	1000	76189580.1999	-125097967.7600	160.9039
R208_100t_20w	3602	760	1078104760.3597	-1042965700.1000	203.3691
R208_25t_5w	320	1000	-4615143.6400	-5060560.5800	8.8017
R208_50t_10w	2826	1000	78354044.4398	-125098091.9200	162.6340
R209_100t_20w	1599	1000	4379948681.1399	-1013243421.1000	532.2701
R209_25t_5w	113	1000	3561388.3599	-5060483.0600	170.3764
R209_50t_10w	642	1000	13855538.1999	-125097999.8800	111.0757
R210_100t_20w	1897	1000	4374544627.5798	-978117773.2200	547.2410
R210_25t_5w	208	1000	-4615260.3600	-5171642.3200	10.7583
R210_50t_10w	810	1000	209514901.4999	-125097934.6800	267.4807
R211_100t_20w	2253	1000	7676388592.9198	-942991911.9800	914.0460
R211_25t_5w	184	1000	-610415.9200	-5060502.8400	87.9376
R211_50t_10w	988	1000	143284902.7398	-122500836.0400	216.9664
RC101_100t_20w	41	1000	30092178382.2198	15814809991.8800	90.2784
RC101_25t_5w	4	1000	39438833.5999	27034405.3800	45.8838
RC101_50t_10w	6	1000	857094402.5400	190034189.6200	351.0211
RC102_100t_20w	63	1000	20935100571.6597	15822916717.4800	32.3087
RC102_25t_5w	4	1000	35322720.5399	26645148.4800	32.5671
RC102_50t_10w	7	1000	781341602.8399	578756222.9600	35.0035
RC103_100t_20w	108	1000	22580618784.7397	17527878844.9600	28.8268
RC103_25t_5w	6	1000	30705762.8999	21805915.8800	40.8139
RC103_50t_10w	13	1000	773117056.3199	767922882.8400	0.6763
RC104_100t_20w	174	1000	20121798982.4197	18130424572.1200	10.9836
RC104_25t_5w	6	1000	26978792.5799	25810879.2400	4.5248
RC104_50t_10w	17	1000	638925988.3199	498674346.2200	28.1248
RC105_100t_20w	62	1000	27519874515.9798	10907978098.4200	152.2912
RC105_25t_5w	4	1000	35100145.5000	30761388.4200	14.1045
RC105_50t_10w	9	1000	982628413.6400	448893536.2400	118.9001
RC106_100t_20w	84	1000	25885164509.9798	11529438265.0600	124.5136
RC106_25t_5w	6	1000	35211517.0399	22417682.5200	57.0702
RC106_50t_10w	9	1000	981762785.9599	515989390.5800	90.2680
RC107_100t_20w	113	1000	29130266772.8798	16638920438.6600	75.0730
RC107_25t_5w	5	1000	30705727.8200	30761589.8800	0.1819 *
RC107_50t_10w	12	1000	773550001.5599	575726019.2200	34.3607
RC108_100t_20w	155	1000	21832164794.1197	14823176358.6800	47.2839
RC108_25t_5w	6	1000	34321529.9399	30372205.5600	13.0030
RC108_50t_10w	16	1000	771818545.4799	630268451.2000	22.4586

Table E.5 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
RC201_100t_20w	460	1000	-1388819673.9802	-1378013035.7800	0.7781 *
RC201_25t_5w	47	1000	-4058811.6000	-5171005.4000	21.5082
RC201_50t_10w	132	1000	-120766859.9402	-125096456.0400	3.4610
RC202_100t_20w	1119	1000	4369142523.4196	-848419159.5200	614.9745
RC202_25t_5w	55	1000	-5171090.3200	-5171480.4200	0.0075
RC202_50t_10w	217	1000	77921726.2198	-125096955.0600	162.2890
RC203_100t_20w	1845	1000	1913023548.4997	-915968950.1800	308.8524
RC203_25t_5w	140	1000	-721067.0600	-5171505.3800	86.0569
RC203_50t_10w	553	1000	77055884.3598	-125096765.1800	161.5970
RC204_100t_20w	3604	982	1907619772.8397	-1032155821.9000	284.8189
RC204_25t_5w	318	1000	-721314.1199	-5060413.3600	85.7459
RC204_50t_10w	1462	1000	76190308.5598	-125097014.9000	160.9049
RC205_100t_20w	998	1000	2739836285.9197	-856525543.8000	419.8779
RC205_25t_5w	72	1000	-276057.1800	-5171462.0600	94.6619
RC205_50t_10w	222	1000	77922090.5797	-125096766.4200	162.2894
RC206_100t_20w	1189	1000	1078106455.3398	-905161565.1000	219.1065
RC206_25t_5w	94	1000	-5171185.8799	-5171397.9400	0.0041
RC206_50t_10w	362	1000	-57999964.4802	-125097137.7600	53.6360
RC207_100t_20w	1422	1000	6852279937.0799	-907864881.6600	854.7686
RC207_25t_5w	97	1000	-721399.1400	-5059890.2000	85.7427
RC207_50t_10w	506	1000	-119035962.7801	-122499526.2000	2.8274
RC208_100t_20w	1856	1000	6857684162.9399	-978116228.2000	801.1113
RC208_25t_5w	141	1000	-888253.6799	-5060744.0000	82.4481
RC208_50t_10w	956	1000	148913526.1999	-125096819.1800	219.0386
test150-0-0-0-0_d0_tw0	3478	1000	101050622678.1997	-28349336446.9000	456.4479
test150-0-0-0-0_d0_tw1	1066	1000	-24141770230.7996	-30832491493.7000	21.7002
test150-0-0-0-0_d0_tw2	809	1000	7242547949.4989	-28832172176.8000	125.1196
test150-0-0-0-0_d0_tw3	628	1000	6897666282.2002	-27383664735.0000	125.1889
test150-0-0-0-0_d0_tw4	915	1000	163819258695.0000	-26142085667.0000	726.6495
test250-0-0-0-0_d0_tw0	-	-	-	23650649218.5000	-
test250-0-0-0-0_d0_tw1	-	-	-	-292254131472.0000	-
test250-0-0-0-0_d0_tw2	3600	823	550721722328.1854	-275698693917.1000	299.7549
test250-0-0-0-0_d0_tw3	3451	1000	-214544933324.4147	-266914175565.0000	19.6202
test250-0-0-0-0_d0_tw4	-	-	-	-231776103927.7000	-
test50-0-0-0-0_d0_tw0	1566	1000	-842598576.7999	-842599324.2000	0.0000
test50-0-0-0-0_d0_tw1	440	1000	-842598443.2999	-842599506.6000	0.0001
test50-0-0-0-0_d0_tw2	331	1000	-842598048.7999	-842599560.3000	0.0001
test50-0-0-0-0_d0_tw3	242	1000	-830114535.8999	-842598590.0000	1.4816
test50-0-0-0-0_d0_tw4	314	1000	-358882360.4999	-842599753.8000	57.4077

E.6 Tabu Search Results: Config. 6

Table E.6: Tabu Search experiments results with parameter configuration 6

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
10_District0	795	1000	71085646155.4499	71563566909.3600	0.6723 *
10_District1	-	-	-	8218857439409.0600	-
10_District2	2438	1000	43508583609.5400	27771011394.0200	56.6690
10_District3	1046	1000	123936186914.5899	139155792160.4600	12.2801 *
10_District4	-	-	-	21291228411410.1000	-
10_District5	-	-	-	37249297856014.8000	-
11_District0	881	1000	3908937400.3999	3046672128.3800	28.3018
11_District1	-	-	-	6157994611789.7000	-
11_District2	1731	1000	13712924670.9100	9214556192.4000	48.8180
11_District3	742	1000	62090125847.7100	61007292651.7800	1.7749
11_District4	-	-	-	22460659385239.4000	-
11_District5	-	-	-	15643005576898.3000	-
12_District0	1165	1000	143925921444.9800	115036288619.9000	25.1134
12_District1	-	-	-	37877356021322.0000	-
12_District2	1007	1000	162025052577.5399	153054550706.1000	5.8609
12_District3	3604	853	242708520028.0901	239836499336.7000	1.1974
12_District4	-	-	-	59129219661289.8000	-
12_District5	-	-	-	65945373929847.0000	-
13_District0	306	1000	181311502302.1600	154315121626.0000	17.4943
13_District1	-	-	-	14674175609787.1000	-
13_District2	2580	1000	116557552317.3498	126837068235.0100	8.8192 *
13_District3	3609	226	530300834518.4200	429665639491.2800	23.4217
13_District4	-	-	-	30033656135618.7000	-
13_District5	-	-	-	42764648834839.3000	-
14_District0	724	1000	36416804859.5300	34977146773.4600	4.1159
14_District1	-	-	-	12434388027267.4000	-
14_District2	1224	1000	108838088822.8200	90165619840.7900	20.7090
14_District3	2155	1000	222228800264.5702	262406322982.9800	18.0793 *
14_District4	-	-	-	31546738861338.4000	-
14_District5	-	-	-	44524141233501.8000	-
15_District0	550	1000	61004148965.9300	42188641727.2300	44.5985
15_District1	-	-	-	12317798430422.9000	-
15_District2	1758	1000	94622695423.7699	67421894826.9300	40.3441
15_District3	3198	1000	387901147155.4305	463823619193.1600	19.5726 *
15_District4	-	-	-	22834329911998.4000	-
15_District5	-	-	-	28700034523627.8000	-
16_District0	339	1000	130510545650.3299	120807470457.0500	8.0318
16_District1	-	-	-	12316160134202.9000	-
16_District2	817	1000	106837284359.2600	94504646913.7700	13.0497
16_District3	3603	563	263833740604.6900	210518442436.7200	25.3257
16_District4	-	-	-	28327769996973.5000	-
16_District5	-	-	-	48522145053303.8000	-

Table E.6 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
17_District0	248	1000	66676523523.1799	60633779564.4100	9.9659
17_District1	-	-	-	12050832937058.4000	-
17_District2	1822	1000	121864222397.2999	111787046906.6000	9.0146
17_District3	1079	1000	213795082745.3099	178730940805.7800	19.6183
17_District4	-	-	-	19092974442252.1000	-
17_District5	-	-	-	23886759047749.6000	-
18_District0	88	1000	2705606036.6999	2341527488.6000	15.5487
18_District1	-	-	-	13924688690304.7000	-
18_District2	1743	1000	6437451587.0199	4417358464.4100	45.7307
18_District3	445	1000	58626690166.1500	70920528499.0700	20.9696 *
18_District4	-	-	-	18718722507656.5000	-
18_District5	-	-	-	12641617614178.6000	-
19_District0	877	1000	47684896990.1200	55462969817.4300	16.3113 *
19_District1	-	-	-	22401449014876.9000	-
19_District2	1997	1000	159584777961.5400	153280470262.9000	4.1129
19_District3	2014	1000	219513932296.3201	243569037679.2000	10.9583 *
19_District4	-	-	-	106192108824987.0000	-
19_District5	-	-	-	78807991381031.8000	-
1_District0	158	199	479423896740.2198	430590936121.1900	11.3409
1_District1	-	-	-	56270206789859.1000	-
1_District2	895	1000	759043419574.2703	672001346398.5800	12.9526
1_District3	3605	333	1186556953800.8809	1031045184732.7300	15.0829
1_District4	-	-	-	73732177837593.4000	-
1_District5	-	-	-	91806570717137.9000	-
20_District0	206	1000	98403745434.5000	98421186256.3800	0.0177 *
20_District1	-	-	-	17222315357479.6000	-
20_District2	1101	567	37915178909.7299	27378465012.0200	38.4854
20_District3	2784	488	309058977399.1803	305213925793.8100	1.2597
20_District4	-	-	-	24960379220572.1000	-
20_District5	-	-	-	48965990159239.8000	-
21_District0	1860	1000	126779461822.4100	120414278382.3300	5.2860
21_District1	-	-	-	16893868904739.4000	-
21_District2	3167	885	99570243545.1099	62666013838.2000	58.8903
21_District3	-	-	-	892017226174.9800	-
21_District4	-	-	-	24304758516183.0000	-
21_District5	-	-	-	45930297567746.2000	-
22_District0	124	1000	155158869527.7698	138906549645.3000	11.7001
22_District1	-	-	-	14071948171020.4000	-
22_District2	1102	1000	234337056419.1298	226837888943.8600	3.3059
22_District3	3600	814	328890339632.4797	285892462254.7700	15.0398
22_District4	-	-	-	29074924558505.3000	-
22_District5	-	-	-	42772519577575.2000	-
23_District0	417	1000	37920541035.5399	32906581132.3800	15.2369
23_District1	-	-	-	17211576588480.4000	-
23_District2	924	1000	221676626879.4699	183871311138.5000	20.5607
23_District3	2169	557	234822540472.9705	250619333021.8100	6.7271 *
23_District4	-	-	-	34555727151866.1000	-
23_District5	-	-	-	70608343796642.5000	-
24_District0	1530	1000	127882819918.6600	105830236823.5000	20.8376
24_District1	-	-	-	11389191027676.9000	-
24_District2	2002	1000	23790717266.3099	26055867878.7800	9.5211 *
24_District3	2965	432	168920805021.7998	159315579521.2200	6.0290
24_District4	-	-	-	24517137738123.8000	-
24_District5	-	-	-	24738655441667.6000	-
25_District0	491	1000	1779384188.4899	1255504790.1000	41.7265
25_District1	-	-	-	6071232244353.8300	-
25_District2	405	1000	3072595371.8500	2430507030.7800	26.4178
25_District3	794	1000	60225323172.5199	54825213021.3900	9.8496
25_District4	-	-	-	20503174454937.0000	-
25_District5	-	-	-	11011719621667.4000	-
26_District0	1857	1000	244312280212.7698	225380467772.9800	8.3999
26_District1	-	-	-	42180173025536.8000	-
26_District2	1069	1000	365344399515.5498	282951685653.1000	29.1190
26_District3	-	-	-	1171533328289.0400	-
26_District4	-	-	-	105580668249051.0000	-
26_District5	-	-	-	76363123397244.6000	-
27_District0	781	1000	162191095687.2999	162479691641.3000	0.1779 *
27_District1	-	-	-	18425601260045.1000	-
27_District2	2671	1000	81249770189.3299	61192189969.0700	32.7780
27_District3	1862	1000	135036770863.5797	116712931257.8400	15.6999
27_District4	-	-	-	33778988955119.3000	-
27_District5	-	-	-	39568513214098.5000	-
28_District0	188	1000	124786919510.0899	102108349093.8000	22.2102
28_District1	-	-	-	15128335139531.3000	-
28_District2	1338	1000	76708481672.4600	64692968024.3600	18.5731
28_District3	1275	404	837122182158.8695	805624097687.0900	3.9097
28_District4	-	-	-	22982825117023.6000	-
28_District5	-	-	-	33791534896023.1000	-
29_District0	1756	1000	33451547170.4700	29353316624.4500	13.9617
29_District1	-	-	-	7484826255380.2500	-
29_District2	2657	1000	83239449171.6500	101233312043.5700	21.6169 *
29_District3	3600	680	289675147038.2499	232601979076.2900	24.5368
29_District4	-	-	-	8309110124637.3200	-
29_District5	-	-	-	35670467010142.6000	-
2_District0	825	1000	132590196057.8599	147005506881.3000	10.8720 *
2_District1	-	-	-	39540340925649.8000	-
2_District2	3601	857	213426529463.1201	217829855689.6200	2.0631 *
2_District3	279	461	972958801223.0298	858787147589.8100	13.2945
2_District4	-	-	-	58876326066126.1000	-
2_District5	-	-	-	95903342380824.9000	-
30_District0	340	1000	15598800537.5799	14856579555.0000	4.9959
30_District1	1512	1000	4672018963434.8390	4288991390631.0100	8.9304
30_District2	988	1000	49059441724.2300	28984654012.0000	69.2600
30_District3	3605	923	113935673232.8501	140260859804.8700	23.1053 *
30_District4	-	-	-	6033193692111.0700	-
30_District5	-	-	-	5491783393822.4800	-
3_District0	340	1000	104223578893.8500	90701681116.3600	14.9081

Table E.6 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
3_District1	-	-	-	17931112587450.7000	-
3_District2	1902	1000	182379150986.7996	184423936337.4900	1.1211 *
3_District3	3601	502	925912592278.8895	837894240881.4400	10.5047
3_District4	-	-	-	67042838268903.4000	-
3_District5	-	-	-	60566955554266.6000	-
4_District0	389	1000	21936168861.7200	20379141775.5100	7.6402
4_District1	-	-	-	16091560843817.3000	-
4_District2	1018	1000	21072131980.1000	19771989271.8700	6.5756
4_District3	1662	1000	33534198365.1000	31243041764.1500	7.3333
4_District4	-	-	-	44497725128783.0000	-
4_District5	-	-	-	30292083288905.3000	-
5_District0	1756	1000	102848687993.0499	105305055950.1000	2.3883 *
5_District1	-	-	-	22578812619408.6000	-
5_District2	1920	1000	73323577261.0801	49746631100.9200	47.3940
5_District3	1136	250	368339564893.9703	366270241695.4900	0.5649
5_District4	-	-	-	68243049339872.1000	-
5_District5	-	-	-	121803986176569.0000	-
6_District0	689	1000	190296879825.6399	174024536017.0000	9.3506
6_District1	-	-	-	24635464198083.6000	-
6_District2	1742	1000	249869266772.6505	263920214770.7000	5.6233 *
6_District3	1938	1000	277503911333.7398	272026860685.1900	2.0134
6_District4	-	-	-	32671893526336.1000	-
6_District5	-	-	-	52577668995018.6000	-
7_District0	1996	1000	50393942153.2799	53992400830.1800	7.1406 *
7_District1	-	-	-	25530776489059.8000	-
7_District2	1230	1000	109291716119.1598	96812988604.0900	12.8895
7_District3	2388	1000	715058901075.4904	627921570619.9200	13.8771
7_District4	-	-	-	30888008083772.9000	-
7_District5	-	-	-	52214576657422.1000	-
8_District0	366	1000	55281651662.3100	47815962594.9100	15.6133
8_District1	-	-	-	13654929607346.6000	-
8_District2	2195	1000	70226429057.8199	73150043166.7900	4.1631 *
8_District3	3401	665	441071185442.4402	367102727288.8900	20.1492
8_District4	-	-	-	29816748603227.9000	-
8_District5	-	-	-	40450402089088.4000	-
9_District0	158	1000	99472150480.8099	100093204504.3000	0.6243 *
9_District1	-	-	-	12178217016824.2000	-
9_District2	690	1000	106870600160.6700	89342594098.5600	19.6188
9_District3	3606	665	1102917837483.7693	911115793606.4100	21.0513
9_District4	-	-	-	37164202673658.1000	-
9_District5	-	-	-	25382936280280.4000	-
C101_100t_20w	3601	832	32950896473.8603	-12304901708.1200	367.7867
C101_25t_5w	148	1000	112140827.5000	9512566.6400	1078.8703
C101_50t_10w	826	1000	268818121.2599	-1079154704.1200	124.9100
C102_100t_20w	3600	590	47541696775.0201	-5155408825.8200	1022.1712
C102_25t_5w	322	1000	77597724.5999	-37546115.7600	306.6731
C102_50t_10w	1198	1000	2025857694.4398	-1102530180.6200	283.7462
C103_100t_20w	611	38	447208023197.6803	-7222438475.8600	6291.9256
C103_25t_5w	1554	1000	78598920.1599	-43052838.4600	282.5638
C103_50t_10w	3222	1000	1955732391.4598	-1110321792.2400	276.1410
C104_100t_20w	279	20	580713841527.5204	-8681519205.2400	6789.0808
C104_25t_5w	2261	1000	113142139.8400	-33540706.8600	437.3278
C104_50t_10w	102	19	17617148100.5599	-1086946666.3200	1720.7923
C105_100t_20w	3600	349	18554641574.9399	-12158993843.3000	252.6001
C105_25t_5w	206	1000	162203263.8600	1002190.7200	16084.8698
C105_50t_10w	2188	1000	245442997.5799	-1102529614.6600	122.2618
C106_100t_20w	3600	244	40076071667.8402	-5349954009.7600	849.0918
C106_25t_5w	237	1000	117647671.6399	9512566.6400	1136.7605
C106_50t_10w	1351	1000	794761334.7799	-1086946589.7000	173.1187
C107_100t_20w	860	47	380455113599.7798	-6979258964.6200	5551.2250
C107_25t_5w	225	1000	120150835.2600	-2001420.3200	6103.2784
C107_50t_10w	3601	976	-409063881.2400	-1125905161.9000	63.6679
C108_100t_20w	447	29	513960932568.2598	-6492899831.1000	8015.7378
C108_25t_5w	459	1000	122653966.8199	-8009072.9800	1631.4377
C108_50t_10w	3605	696	179212996.2200	-1125905177.3200	115.9172
C109_100t_20w	459	24	551045881748.0600	-6614488230.8200	8430.8921
C109_25t_5w	361	1000	39049721.0799	-26031566.9600	250.0091
C109_50t_10w	3607	477	-455814576.5400	-1125905761.1800	59.5157
C201_100t_20w	569	34	476875982470.3000	-12596720162.2300	3885.7154
C201_25t_5w	3278	1000	-42552140.6200	-45555916.9200	6.5936
C201_50t_10w	3608	215	-506460875.0600	-1125905241.3800	55.0174
C202_100t_20w	105	7	677134710773.2001	-6055175633.1600	11282.7426
C202_25t_5w	1031	76	-42552011.0800	-45555950.8800	6.5939
C202_50t_10w	108	20	16974329031.3799	-1125905361.9600	1607.6159
C203_100t_20w	-	-	-	-6930623788.8800	-
C203_25t_5w	7	9	576720472.3799	-45555956.2400	1365.9606
C203_50t_10w	67	12	21824693208.1399	-1125905456.0600	2038.4125
C204_100t_20w	-	-	-	-8657198950.2400	-
C204_25t_5w	8	7	656820424.1599	-45555977.0000	1541.7875
C204_50t_10w	108	12	21824693208.1399	-1125905585.1400	2038.4123
C205_100t_20w	250	16	610381801286.5201	-12596718687.8400	4945.5618
C205_25t_5w	3606	470	-42552064.5799	-45555926.8000	6.5937
C205_50t_10w	462	32	9728002118.8399	-1125905403.9800	964.0159
C206_100t_20w	191	12	640049760925.9800	-7733117210.9600	8376.7368
C206_25t_5w	3602	371	-38547010.7400	-45555926.8000	15.3853
C206_50t_10w	229	21	16370468487.2999	-1125905541.4200	1553.9824
C207_100t_20w	162	12	640049760925.9800	-7222438653.2400	8961.9618
C207_25t_5w	149	13	416520771.4199	-45555933.4200	1014.3063
C207_50t_10w	131	15	20032590733.9399	-1125905455.9000	1879.2427
C208_100t_20w	145	9	662300730815.0200	-8657201177.3800	7750.2869
C208_25t_5w	123	11	496620584.9799	-45555933.1600	1190.1337
C208_50t_10w	198	18	18201529571.0599	-1125905465.6600	1716.6125
hh_00_P0	3602	868	147857350548.1860	6663651008.2900	2118.8639
ll1_00_P0	471	1000	3750702178.7599	1338732826.5000	180.1680
ll1_01_P0	310	1000	7003162322.1899	1338732826.5000	423.1187
ll1_02_P0	489	1000	3719565458.8901	1338732826.5000	177.8422
ll1_03_P0	282	1000	3688486937.4401	1338732826.5000	175.5207

Table E.6 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
ll1_04_P0	422	1000	3812955950.4797	1338732826.5000	184.8182
ll1_05_P0	410	1000	8730593841.5700	1307660206.1200	567.6500
ll1_06_P0	401	1000	7034329424.1001	1214281948.4400	479.2995
ll1_07_P0	570	1000	2132222396.3500	1338732826.5000	59.2716
ll2_00_P0	754	1000	-179546894.9401	85352916.2700	147.5379 *
ll3_00_P0	837	1000	-179677818.6500	-198615845.6200	9.5350
R101_100t_20w	31	1000	14685373535.4997	6319981502.8600	132.3641
R101_25t_5w	4	1000	60631864.9800	47893886.5800	26.5962
R101_50t_10w	12	1000	1241054540.6999	709051142.3992	75.0303
R102_100t_20w	88	1000	15339257933.8198	14101742003.1800	8.7756
R102_25t_5w	6	1000	38715883.0199	34655177.0000	11.7174
R102_50t_10w	22	1000	1096907396.5798	641955625.9388	70.8696
R103_100t_20w	148	1000	10408108040.2398	12469734030.7400	19.8078 *
R103_25t_5w	25	1000	30705744.8800	30260803.3400	1.4703
R103_50t_10w	85	1000	831122334.4799	570531480.7600	45.6751
R104_100t_20w	485	1000	11953652468.3598	14023383931.9600	17.3146 *
R104_25t_5w	24	1000	30205265.2599	29982663.2800	0.7424
R104_50t_10w	176	1000	448893633.9799	293491458.8800	52.9494
R105_100t_20w	170	1000	10364875724.4598	8424839701.0800	23.0275
R105_25t_5w	11	1000	47782625.1399	35155829.6400	35.9166
R105_50t_10w	28	1000	900382043.5200	388290958.0600	131.8833
R106_100t_20w	151	1000	11234919995.9198	8443754034.7200	33.0559
R106_25t_5w	8	1000	34210195.3400	30483330.4600	12.2259
R106_50t_10w	29	1000	764892234.7197	303880567.4400	151.7081
R107_100t_20w	263	1000	11991480297.6198	11634816307.3400	3.0654
R107_25t_5w	20	1000	34265820.8599	22028382.6800	55.5530
R107_50t_10w	90	1000	635895640.9399	372274723.4800	70.8135
R108_100t_20w	390	1000	7117072275.7998	12369760088.0800	73.8040 *
R108_25t_5w	17	1000	30093948.4000	26033406.5200	15.5974
R108_50t_10w	80	1000	442833495.6999	293491505.9600	50.8846
R109_100t_20w	347	1000	7938479894.3598	6636116061.1200	19.6253
R109_25t_5w	9	1000	42887736.4199	26255954.0600	63.3448
R109_50t_10w	41	1000	841944062.2198	631133985.4200	33.4017
R110_100t_20w	377	1000	12839908120.0998	7622345982.8200	68.4508
R110_25t_5w	6	1000	43332769.6199	34488424.6600	25.6443
R110_50t_10w	29	1000	515556396.5195	636761297.5800	23.5095 *
R111_100t_20w	178	1000	9694780262.8597	8473476077.2600	14.4132
R111_25t_5w	9	1000	38382037.9600	26144634.7600	46.8065
R111_50t_10w	45	1000	510794689.0597	496942686.1600	2.7874
R112_100t_20w	484	1000	11210602285.8198	10851236130.0600	3.3117
R112_25t_5w	7	1000	26478392.1799	26311467.9000	0.6344
R112_50t_10w	24	1000	571397274.2998	427249792.1800	33.7384
R201_100t_20w	3605	278	253994823.6798	-1383419122.5000	118.3599
R201_25t_5w	762	1000	-5004446.8401	-5171519.4600	3.2306
R201_50t_10w	3611	354	-124231486.3201	-125097475.6400	0.6922
R202_100t_20w	549	39	48865673126.6398	-778169509.0800	6379.5666
R202_25t_5w	1481	1000	-4559486.3799	-5171550.1800	11.8352
R202_50t_10w	3604	741	143285314.5799	-125097689.9000	214.5387
R203_100t_20w	673	37	50513893127.7998	-891653653.5400	5765.1921
R203_25t_5w	3047	1000	-387777.3000	-5171614.5200	92.5018
R203_50t_10w	347	30	1217247248.6999	-125097805.5400	1073.0364
R204_100t_20w	329	20	64523761782.2999	-1007840112.0800	6502.1823
R204_25t_5w	58	6	77430354.5400	-5060450.1000	1630.1080
R204_50t_10w	101	19	1955298539.0799	-125098025.1800	1663.0131
R205_100t_20w	3623	136	1896810807.6597	-910567724.8000	308.3107
R205_25t_5w	1809	1000	-721670.6201	-5171759.2000	86.0459
R205_50t_10w	3606	276	9094043.7598	-125097876.3800	107.2695
R206_100t_20w	554	30	56282662602.6398	-964607487.1800	5934.7735
R206_25t_5w	2754	1000	-721159.3600	-5171813.9000	86.0559
R206_50t_10w	3601	303	77055451.4998	-125097837.4400	161.5961
R207_100t_20w	688	30	56282662503.9198	-1018647805.1000	5625.2327
R207_25t_5w	3602	941	-554660.5200	-5171794.9600	89.2752
R207_50t_10w	1254	251	76189580.1999	-125097967.7600	160.9039
R208_100t_20w	385	20	64523761848.9599	-1042965700.1000	6286.5660
R208_25t_5w	58	6	77430354.5400	-5060560.5800	1630.0746
R208_50t_10w	188	17	2091654106.6199	-125098091.9200	1772.0111
R209_100t_20w	826	43	45569233634.9198	-1013243421.1000	4597.3628
R209_25t_5w	3284	1000	-220852.6200	-5060483.0600	95.6357
R209_50t_10w	3600	265	13855498.4399	-125097999.8800	111.0757
R210_100t_20w	502	37	50513893110.3798	-978117773.2200	5264.3978
R210_25t_5w	2565	1000	-554605.7200	-5171642.3200	89.2760
R210_50t_10w	3605	251	209514901.4999	-125097934.6800	267.4807
R211_100t_20w	496	30	56282662891.2198	-942991911.9800	6068.5202
R211_25t_5w	2703	1000	-609909.4400	-5060502.8400	87.9476
R211_50t_10w	499	30	1217247406.4399	-122500836.0400	1093.6645
RC101_100t_20w	118	1000	16268746076.4196	15814809991.8800	2.8703
RC101_25t_5w	9	1000	39438826.4000	27034405.3800	45.8838
RC101_50t_10w	25	1000	912502825.8399	190034189.6200	380.1782
RC102_100t_20w	104	1000	10548612288.4796	15822916717.4800	49.9999 *
RC102_25t_5w	18	1000	26645156.2800	26645148.4800	0.0000
RC102_50t_10w	38	1000	716410393.7799	578756222.9600	23.7844
RC103_100t_20w	157	1000	8805822215.7195	17527878844.9600	99.0487 *
RC103_25t_5w	10	1000	26422790.7200	21805915.8800	21.1725
RC103_50t_10w	32	1000	519885192.7600	767922882.8400	47.7100 *
RC104_100t_20w	363	1000	8016838615.5197	18130424572.1200	126.1542 *
RC104_25t_5w	18	1000	25977712.6400	25810879.2400	0.6463
RC104_50t_10w	79	1000	708618887.2199	498674346.2200	42.1005
RC105_100t_20w	60	1000	10502678165.0194	10907978098.4200	3.8590 *
RC105_25t_5w	12	1000	35100122.5800	30761388.4200	14.1044
RC105_50t_10w	59	1000	985658673.0599	448893536.2400	119.5751
RC106_100t_20w	200	1000	12104964337.2395	11529438265.0600	4.9917
RC106_25t_5w	30	1000	39160712.3998	22417682.5200	74.6867
RC106_50t_10w	30	1000	583517762.9399	515989390.5800	13.0871
RC107_100t_20w	342	1000	10567526143.5597	16638920438.6600	57.4533 *
RC107_25t_5w	4	1000	30817076.0599	30761589.8800	0.1803
RC107_50t_10w	26	1000	511227747.8200	575726019.2200	12.6163 *
RC108_100t_20w	216	1000	10608056456.5398	14823176358.6800	39.7350 *

Table E.6 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
RC108_25t_5w	6	1000	26533934.6198	30372205.5600	14.4655 *
RC108_50t_10w	33	1000	583950807.2999	630268451.2000	7.9317 *
RC201_100t_20w	3604	262	-1388819673.9802	-1378013035.7800	0.7781 *
RC201_25t_5w	464	1000	-4225232.3401	-5171005.4000	18.2899
RC201_50t_10w	3603	617	-121199307.3601	-125096456.0400	3.1153
RC202_100t_20w	3604	401	4369142523.4196	-848419159.5200	614.9745
RC202_25t_5w	856	1000	-721020.8200	-5171480.4200	86.0577
RC202_50t_10w	3601	765	14290195.8399	-125096955.0600	111.4232
RC203_100t_20w	2816	302	1913023548.4997	-915968950.1800	308.8524
RC203_25t_5w	2379	1000	-387464.4800	-5171505.3800	92.5077
RC203_50t_10w	3605	497	77055884.3598	-125096765.1800	161.5970
RC204_100t_20w	454	26	59579102992.6399	-1032155821.9000	5872.2973
RC204_25t_5w	3605	836	4062399.6399	-5060413.3600	180.2780
RC204_50t_10w	233	25	1554889523.9399	-125097014.9000	1342.9469
RC205_100t_20w	3611	184	2739836285.9197	-856525543.8000	419.8779
RC205_25t_5w	928	1000	-4336930.4400	-5171462.0600	16.1372
RC205_50t_10w	3602	606	77922083.9997	-125096766.4200	162.2894
RC206_100t_20w	3602	168	1078106455.3398	-905161565.1000	219.1065
RC206_25t_5w	1120	1000	-554129.1000	-5171397.9400	89.2847
RC206_50t_10w	3601	456	-119035754.7601	-125097137.7600	4.8453
RC207_100t_20w	745	41	47217454544.6000	-907864881.6600	5300.9341
RC207_25t_5w	2759	1000	-4281054.2400	-5059890.2000	15.3923
RC207_50t_10w	3103	453	-119035962.7801	-122499526.2000	2.8274
RC208_100t_20w	469	35	52162114161.3600	-978116228.2000	5432.9157
RC208_25t_5w	1466	1000	-4781935.9400	-5060744.0000	5.5092
RC208_50t_10w	395	29	1284343085.1399	-125096819.1800	1126.6792
test150-0-0-0-0_d0_tw0	-	-	-	-28349336446.9000	-
test150-0-0-0-0_d0_tw1	-	-	-	-30832491493.7000	-
test150-0-0-0-0_d0_tw2	-	-	-	-28832172176.8000	-
test150-0-0-0-0_d0_tw3	-	-	-	-27383664735.0000	-
test150-0-0-0-0_d0_tw4	-	-	-	-26142085667.0000	-
test250-0-0-0-0_d0_tw0	-	-	-	23650649218.5000	-
test250-0-0-0-0_d0_tw1	-	-	-	-292254131472.0000	-
test250-0-0-0-0_d0_tw2	-	-	-	-275698693917.1000	-
test250-0-0-0-0_d0_tw3	-	-	-	-266914175565.0000	-
test250-0-0-0-0_d0_tw4	-	-	-	-231776103927.7000	-
test50-0-0-0-0_d0_tw0	183	14	16571183879.9000	-842599324.2000	2066.6742
test50-0-0-0-0_d0_tw1	3600	660	-842598568.6999	-842599506.6000	0.0001
test50-0-0-0-0_d0_tw2	3600	868	-842598048.7999	-842599560.3000	0.0001
test50-0-0-0-0_d0_tw3	3600	827	-836356393.4000	-842598590.0000	0.7408
test50-0-0-0-0_d0_tw4	3600	819	-358882360.4999	-842599753.8000	57.4077

E.7 Tabu Search Results: Config. 7

Table E.7: Tabu Search experiments results with parameter configuration 7

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
10_District0	83	1000	90825886061.7299	71563566909.3600	26.9163
10_District1	963	1000	10391910011489.5490	8218857439409.0600	26.4398
10_District2	201	1000	62409687842.2898	27771011394.0200	124.7296
10_District3	125	1000	173382042749.8000	139155792160.4600	24.5956
10_District4	3363	1000	28824022749673.9530	21291228411410.1000	35.3798
10_District5	3085	1000	42657438984971.2600	37249297856014.8000	14.5187
11_District0	46	1000	4557233152.3900	3046672128.3800	49.5806
11_District1	804	1000	7519784104587.9960	6157994611789.7000	22.1141
11_District2	46	1000	20020110812.9300	9214556192.4000	117.2661
11_District3	103	1000	99044691607.9400	61007292651.7800	62.3489
11_District4	2849	1000	28327341544190.6520	22460659385239.4000	26.1198
11_District5	1219	1000	17354107747842.5300	15643005576898.3000	10.9384
12_District0	29	1000	160068033926.6700	115036288619.9000	39.1456
12_District1	2168	1000	46524033514596.0800	37877356021322.0000	22.8280
12_District2	97	1000	231564970965.7198	153054550706.1000	51.2957
12_District3	161	1000	281412413711.2700	239836499336.7000	17.3351
12_District4	3600	596	69976301190337.2660	59129219661289.8000	18.3447
12_District5	2634	1000	77604572919275.8100	65945373929847.0000	17.6800
13_District0	30	1000	185143085164.5900	154315121626.0000	19.9772
13_District1	1647	1000	19871772650002.9260	14674175609787.1000	35.4200
13_District2	149	1000	154666155183.3599	126837068235.0100	21.9408
13_District3	254	1000	545562728441.2301	429665639491.2800	26.9737
13_District4	3600	792	34944914457256.2730	30033656135618.7000	16.3525
13_District5	3602	750	56929657841132.4100	42764648834839.3000	33.1231
14_District0	22	1000	39405185830.6300	34977146773.4600	12.6598
14_District1	2061	1000	14524429777197.5840	12434388027267.4000	16.8085
14_District2	88	1000	148274342935.6299	90165619840.7900	64.4466
14_District3	226	1000	318815165103.5098	262406322982.9800	21.4967
14_District4	3601	849	43274108721617.1600	31546738861338.4000	37.1745
14_District5	3583	1000	51310668743104.5400	44524141233501.8000	15.2423
15_District0	35	1000	67330404477.5899	42188641727.2300	59.5936
15_District1	859	1000	14069935349289.4470	12317798430422.9000	14.2244
15_District2	151	1000	103156785657.2899	67421894826.9300	53.0019
15_District3	270	1000	452400421567.4308	463823619193.1600	2.5250 *
15_District4	3349	1000	32194102455076.9380	22834329911998.4000	40.9899
15_District5	2776	1000	33961376146576.4570	28700034523627.8000	18.3321
16_District0	20	1000	133772786389.0099	120807470457.0500	10.7322
16_District1	1566	1000	16792011684544.6100	12316160134202.9000	36.3412
16_District2	69	1000	129096190289.2900	94504646913.7700	36.6030
16_District3	141	1000	273024373859.2299	210518442436.7200	29.6914

Table E.7 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
16_District4	3603	832	35952100828187.3600	28327769996973.5000	26.9146
16_District5	3014	1000	54102712144284.7300	48522145053303.8000	11.5010
17_District0	14	1000	71039090116.1100	60633779564.4100	17.1609
17_District1	1064	1000	13583226327478.4280	12050832937058.4000	12.7160
17_District2	119	1000	147556479491.0699	111787046906.6000	31.9978
17_District3	134	1000	263855484278.8798	178730940805.7800	47.6272
17_District4	3600	836	22294458031455.2340	19092974442252.1000	16.7678
17_District5	2206	1000	28598180584785.0620	23886759047749.6000	19.7239
18_District0	5	1000	3074444221.2799	2341527488.6000	31.3007
18_District1	1176	1000	15347660439622.5760	13924688690304.7000	10.2190
18_District2	46	1000	8988138806.4500	4417358464.4100	103.4731
18_District3	96	1000	76037039331.1500	70920528499.0700	7.2144
18_District4	3282	1000	21449713637219.9300	18718722507656.5000	14.5896
18_District5	1596	1000	15543852583612.3090	12641617614178.6000	22.9577
19_District0	39	1000	56494154145.1400	55462969817.4300	1.8592
19_District1	2450	1000	26413497392708.1800	22401449014876.9000	17.9097
19_District2	101	1000	201290195807.8100	153280470262.9000	31.3214
19_District3	202	1000	274547752181.1901	243569037679.2000	12.7186
19_District4	-	-	-	106192108824987.0000	-
19_District5	3473	1000	92374621349656.3100	78807991381031.8000	17.2147
1_District0	81	1000	512177107903.6697	430590936121.1900	18.9474
1_District1	3601	969	64113401266731.5800	56270206789859.1000	13.9384
1_District2	191	1000	895752627625.0302	672001346398.5800	33.2962
1_District3	342	1000	1245277165331.9802	1031045184732.7300	20.7781
1_District4	3606	393	89000469461106.8300	73732177837593.4000	20.7077
1_District5	3601	660	105190510015786.4700	91806570717137.9000	14.5784
20_District0	17	1000	104403548586.2000	98421186256.3800	6.0783
20_District1	1742	1000	20166473748877.4140	17222315357479.6000	17.0950
20_District2	147	1000	49913437019.9999	27378465012.0200	82.3091
20_District3	249	1000	332285164824.6803	305213925793.8100	8.8695
20_District4	3602	844	30556157284792.6700	24960379220572.1000	22.4186
20_District5	3364	1000	54096789514340.4500	48965990159239.8000	10.4782
21_District0	59	1000	153837727366.3701	120414278382.3300	27.7570
21_District1	1826	1000	20356277402674.2970	16893868904739.4000	20.4950
21_District2	114	1000	108413022554.9099	62666013838.2000	73.0013
21_District3	608	1000	1083818307996.6499	892017226174.9800	21.5019
21_District4	3604	838	27599158275505.6500	24304758516183.0000	13.5545
21_District5	3604	926	57789774017343.4800	45930297567746.2000	25.8205
22_District0	28	1000	164148433829.6999	138906549645.3000	18.1718
22_District1	1860	1000	17528245590773.4550	14071948171020.4000	24.5616
22_District2	124	1000	345487141017.4400	226837888943.8600	52.3057
22_District3	172	1000	389586838914.7998	285892462254.7700	36.2704
22_District4	3602	822	36430067637167.2660	29074924558505.3000	25.2972
22_District5	3601	921	49609639676068.9840	42772519577575.2000	15.9848
23_District0	11	1000	35354341384.0000	32906581132.3800	7.4385
23_District1	1890	1000	20306570703942.8700	17211576588480.4000	17.9820
23_District2	89	1000	252485503754.2099	183871311138.5000	37.3164
23_District3	210	1000	286075456109.0404	250619333021.8100	14.1474
23_District4	3601	562	44124735609381.3750	34555727151866.1000	27.6915
23_District5	3602	808	82318311632011.2200	70608343796642.5000	16.5843
24_District0	43	1000	139341297826.3999	105830236823.5000	31.6649
24_District1	1105	1000	14601551656495.2070	11389191027676.9000	28.2053
24_District2	70	1000	27287793037.5799	26055867878.7800	4.7280
24_District3	158	1000	176442969278.3299	159315579521.2200	10.7506
24_District4	3600	829	28492637314463.2800	24517137738123.8000	16.2151
24_District5	2325	1000	29285060431772.2460	24738655441667.6000	18.3777
25_District0	14	1000	1797448948.6899	1255504790.1000	43.1654
25_District1	889	1000	7110320772712.6380	6071232244353.8300	17.1149
25_District2	12	1000	3671152394.0500	2430507030.7800	51.0447
25_District3	74	1000	71806281069.8399	54825213021.3900	30.9730
25_District4	3602	991	25996861803541.5400	20503174454937.0000	26.7943
25_District5	897	1000	12406452080578.2800	11011719621667.4000	12.6658
26_District0	72	1000	273316551246.4900	225380467772.9800	21.2689
26_District1	2772	1000	48987132161122.1500	42180173025536.8000	16.1378
26_District2	120	1000	423888084465.3799	282951685653.1000	49.8093
26_District3	629	1000	1292287659214.3389	1171533328289.0400	10.3073
26_District4	3601	440	126120081289807.2700	105580668249051.0000	19.4537
26_District5	3601	619	88272702517243.4700	76363123397244.6000	15.5959
27_District0	48	1000	203950972477.4898	162479691641.3000	25.5239
27_District1	1568	1000	24579099095816.6900	18425601260045.1000	33.3964
27_District2	162	1000	107092699865.6099	61192189969.0700	75.0104
27_District3	111	1000	140452684201.0099	116712931257.8400	20.3402
27_District4	3606	584	42830700078064.5100	33778988955119.3000	26.7968
27_District5	2644	1000	46799178020401.0300	39568513214098.5000	18.2737
28_District0	24	1000	139029500086.9099	102108349093.8000	36.1587
28_District1	1700	1000	16751294190753.8380	15128335139531.3000	10.7279
28_District2	64	1000	100507447531.4600	64692968024.3600	55.3606
28_District3	464	1000	957802714175.6798	805624097687.0900	18.8895
28_District4	3134	1000	28571951497097.7540	22982825117023.6000	24.3187
28_District5	2864	1000	41410160521065.4840	33791534896023.1000	22.5459
29_District0	25	1000	37601785686.8100	29353316624.4500	28.1006
29_District1	1594	1000	9019729753026.5980	7484826255380.2500	20.5068
29_District2	116	1000	108362419182.7700	101233312043.5700	7.0422
29_District3	181	1000	289675147038.2499	232601979076.2900	24.5368
29_District4	2806	1000	10869563929843.5680	8309110124637.3200	30.8150
29_District5	3164	1000	40098034357696.8600	35670467010142.6000	12.4124
2_District0	88	1000	193839933442.6700	147005506881.3000	31.8589
2_District1	3004	1000	45459857429067.4450	39540340925649.8000	14.9708
2_District2	141	1000	257280064254.1997	217829855689.6200	18.1105
2_District3	321	1000	1210994966019.4000	858787147589.8100	41.0122
2_District4	3606	458	74239030120333.2500	58876326066126.1000	26.0931
2_District5	3600	640	113193541593170.9200	95903342380824.9000	18.0287
30_District0	11	1000	16973733041.8800	14856579555.0000	14.2506
30_District1	533	1000	5453874315157.2070	4288991390631.0100	27.1598
30_District2	88	1000	51330273875.4600	28984654012.0000	77.0946
30_District3	115	1000	171598828550.1400	140260859804.8700	22.3426
30_District4	1582	1000	6957178114976.4290	6033193692111.0700	15.3150

Table E.7 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
30_District5	682	1000	6652589252190.8170	5491783393822.4800	21.1371
3_District0	37	1000	114637704174.6800	90701681116.3600	26.3898
3_District1	1583	1000	21669256497445.5620	17931112587450.7000	20.8472
3_District2	149	1000	226538655017.0398	184423936337.4900	22.8358
3_District3	355	1000	1055208515491.4495	837894240881.4400	25.9357
3_District4	-	-	-	67042838268903.4000	-
3_District5	3005	1000	71450369212445.9400	60566955554266.6000	17.9692
4_District0	13	1000	22783577331.0199	20379141775.5100	11.7985
4_District1	988	1000	18643409643794.8630	16091560843817.3000	15.8583
4_District2	49	1000	28590883223.4399	19771989271.8700	44.6029
4_District3	71	1000	44008054917.4100	31243041764.1500	40.8571
4_District4	3605	531	55827883212635.5600	44497725128783.0000	25.4623
4_District5	2074	1000	35485732192312.5100	30292083288905.3000	17.1452
5_District0	34	1000	121337075496.5599	105305055950.1000	15.2243
5_District1	2815	1000	26958729717452.4450	22578812619408.6000	19.3983
5_District2	79	1000	84445692130.8600	49746631100.9200	69.7515
5_District3	231	1000	368339564893.9703	366270241695.4900	0.5649
5_District4	3607	434	87052360312381.4500	68243049339872.1000	27.5622
5_District5	3603	540	144263825380312.4000	121803986176569.0000	18.4393
6_District0	45	1000	229881593831.3499	174024536017.0000	32.0972
6_District1	3371	1000	29757855339023.4100	24635464198083.6000	20.7927
6_District2	122	1000	314699930704.3200	263920214770.7000	19.2405
6_District3	172	1000	326846711901.7501	272026860685.1900	20.1523
6_District4	3601	849	41017633874659.3900	32671893526336.1000	25.5440
6_District5	3455	1000	65047160187885.1250	52577668995018.6000	23.7163
7_District0	55	1000	64167354628.6999	53992400830.1800	18.8451
7_District1	1896	1000	28469891691201.0230	25530776489059.8000	11.5120
7_District2	82	1000	131369464511.8198	96812988604.0900	35.6940
7_District3	405	1000	885810255449.4302	627921570619.9200	41.0702
7_District4	3602	787	38983888110125.7900	30888008083772.9000	26.2104
7_District5	3603	797	59599457455793.6500	52214576657422.1000	14.1433
8_District0	28	1000	66981129846.9999	47815962594.9100	40.0811
8_District1	1426	1000	15922201158605.7680	13654929607346.6000	16.6040
8_District2	65	1000	92183374224.6500	73150043166.7900	26.0195
8_District3	322	1000	482102510681.2601	367102727288.8900	31.3263
8_District4	3600	929	35399491028950.2900	29816748603227.9000	18.7235
8_District5	3026	1000	50474401568700.1000	40450402089088.4000	24.7809
9_District0	17	1000	115412536479.3100	100093204504.3000	15.3050
9_District1	1199	1000	15318565989602.6800	12178217016824.2000	25.7866
9_District2	52	1000	142708115059.6600	89342594098.5600	59.7313
9_District3	430	1000	1284512871128.3906	911115793606.4100	40.9823
9_District4	3608	552	48344285222867.3100	37164202673658.1000	30.0829
9_District5	2649	1000	32702225945411.4060	25382936280280.4000	28.8354
C101_100t_20w	72	1000	99095856585.7203	-12304901708.1200	905.3364
C101_25t_5w	6	1000	155194466.4599	9512566.6400	1531.4678
C101_50t_10w	20	1000	2610238793.0599	-1079154704.1200	341.8780
C102_100t_20w	276	1000	98901313024.3402	-5155408825.8200	2018.3990
C102_25t_5w	17	1000	190738933.2400	-37546115.7600	608.0124
C102_50t_10w	70	1000	3140078105.8398	-1102530180.6200	384.8065
C103_100t_20w	507	1000	39930163004.1205	-7222438475.8600	652.8626
C103_25t_5w	63	1000	189237129.5600	-43052838.4600	539.5462
C103_50t_10w	149	1000	3743938477.6599	-1110321792.2400	437.1940
C104_100t_20w	768	1000	17314423586.3605	-8681519205.2400	299.4400
C104_25t_5w	106	1000	192240902.4600	-33540706.8600	673.1569
C104_50t_10w	251	1000	1305120952.8799	-1086946666.3200	220.0722
C105_100t_20w	130	1000	77185339028.9999	-12158993843.3000	734.8003
C105_25t_5w	9	1000	196746350.4799	1002190.7200	19531.6276
C105_50t_10w	33	1000	3186828351.9399	-1102529614.6600	389.0469
C106_100t_20w	169	1000	54739825368.8202	-5349954009.7600	1123.1831
C106_25t_5w	8	1000	232791280.2999	9512566.6400	2347.1973
C106_50t_10w	27	1000	1379142584.3799	-1086946589.7000	226.8822
C107_100t_20w	194	1000	69525168735.1598	-6979258964.6200	1096.1683
C107_25t_5w	10	1000	191740192.8600	-2001420.3200	9680.2061
C107_50t_10w	46	1000	3747834654.6599	-1125905161.9000	432.8730
C108_100t_20w	269	1000	39735619043.3798	-6492899831.1000	711.9857
C108_25t_5w	18	1000	193242032.7000	-8009072.9800	2512.7890
C108_50t_10w	61	1000	2551800875.0799	-1125905177.3200	326.6443
C109_100t_20w	394	1000	47006701457.3999	-6614488230.8200	810.6627
C109_25t_5w	18	1000	151189574.8799	-26031566.9600	680.7932
C109_50t_10w	87	1000	2524529654.7199	-1125905761.1800	324.2221
C201_100t_20w	249	1000	-12596716053.9600	-12596720162.2300	0.0000
C201_25t_5w	18	1000	-42552000.2200	-45555916.9200	6.5939
C201_50t_10w	54	1000	-506460875.0600	-1125905241.3800	55.0174
C202_100t_20w	1128	1000	39322214142.4203	-6055175633.1600	749.3984
C202_25t_5w	95	1000	-42552011.0800	-45555950.8800	6.5939
C202_50t_10w	243	1000	685677219.4399	-1125905361.9600	160.9000
C203_100t_20w	2489	1000	24536869879.3203	-6930623788.8800	454.0355
C203_25t_5w	1338	1000	-2001330.3400	-45555956.2400	95.6068
C203_50t_10w	583	1000	709052735.5999	-1125905456.0600	162.9762
C204_100t_20w	3613	812	24609823628.3203	-8657198950.2400	384.2700
C204_25t_5w	1575	1000	-5005102.0200	-45555977.0000	89.0132
C204_50t_10w	1409	1000	93504394.2998	-1125905585.1400	108.3048
C205_100t_20w	511	1000	9702889586.1598	-12596718687.8400	177.0271
C205_25t_5w	40	1000	-42552036.5600	-45555926.8000	6.5938
C205_50t_10w	116	1000	-522044220.5000	-1125905403.9800	53.6333
C206_100t_20w	872	1000	31953859516.5598	-7733117210.9600	513.2080
C206_25t_5w	66	1000	-5005181.3000	-45555926.8000	89.0131
C206_50t_10w	168	1000	677885628.8399	-1125905541.4200	160.2080
C207_100t_20w	1235	1000	24488233802.3396	-7222438653.2400	439.0576
C207_25t_5w	119	1000	32541792.4999	-45555933.4200	171.4326
C207_50t_10w	368	1000	-498668527.7600	-1125905455.9000	55.7095
C208_100t_20w	1203	1000	24536869467.2000	-8657201177.3800	383.4272
C208_25t_5w	113	1000	-41050140.9600	-45555933.1600	9.8906
C208_50t_10w	249	1000	697364998.0999	-1125905465.6600	161.9381
hh_00_P0	171	1000	855509231017.3870	6663651008.2900	12738.4459
ll1_00_P0	46	1000	10395714853.6900	1338732826.5000	676.5339
ll1_01_P0	45	1000	8761695462.9999	1338732826.5000	554.4767

Table E.7 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
ll1_02_P0	44	1000	8761707865.2201	1338732826.5000	554.4777
ll1_03_P0	45	1000	10364606753.6001	1338732826.5000	674.2102
ll1_04_P0	46	1000	10333442162.1100	1338732826.5000	671.8823
ll1_05_P0	46	1000	15360038245.1100	1307660206.1200	1074.6199
ll1_06_P0	45	1000	12029720714.7801	1214281948.4400	890.6859
ll1_07_P0	46	1000	5431391939.7000	1338732826.5000	305.7114
ll2_00_P0	28	1000	402249701.6198	85352916.2700	371.2782
ll3_00_P0	26	1000	965681078.0999	-198615845.6200	586.2054
R101_100t_20w	19	1000	28343983676.1399	6319981502.8600	348.4820
R101_25t_5w	3	1000	56181910.1599	47893886.5800	17.3049
R101_50t_10w	5	1000	1697304701.0199	709051142.3992	139.3769
R102_100t_20w	48	1000	20929696190.8198	14101742003.1800	48.4192
R102_25t_5w	3	1000	38715883.0199	34655177.0000	11.7174
R102_50t_10w	12	1000	1291268163.0199	641955625.9388	101.1460
R103_100t_20w	74	1000	20094778348.1398	12469734030.7400	61.1484
R103_25t_5w	6	1000	39383308.5600	30260803.3400	30.1462
R103_50t_10w	19	1000	1095176025.1799	570531480.7600	91.9571
R104_100t_20w	105	1000	19294986670.3198	14023383931.9600	37.5915
R104_25t_5w	8	1000	38159630.0000	29982663.2800	27.2723
R104_50t_10w	24	1000	1025916190.6799	293491458.8800	249.5557
R105_100t_20w	33	1000	24201817987.1599	8424839701.0800	187.2674
R105_25t_5w	4	1000	47782554.3599	35155829.6400	35.9164
R105_50t_10w	7	1000	1628044920.3799	388290958.0600	319.2847
R106_100t_20w	59	1000	24163990277.4198	8443754034.7200	186.1759
R106_25t_5w	4	1000	42998863.9400	30483330.4600	41.0569
R106_50t_10w	12	1000	1422862326.5399	303880567.4400	368.2307
R107_100t_20w	80	1000	20113692439.8597	11634816307.3400	72.8750
R107_25t_5w	6	1000	42609535.5800	22028382.6800	93.4301
R107_50t_10w	17	1000	1028080394.0399	372274723.4800	176.1617
R108_100t_20w	104	1000	17654872808.1198	12369760088.0800	42.7260
R108_25t_5w	7	1000	38159630.0000	26033406.5200	46.5794
R108_50t_10w	24	1000	1089981863.1799	293491505.9600	271.3844
R109_100t_20w	57	1000	22548193919.5199	6636116061.1200	239.7799
R109_25t_5w	4	1000	42776380.8200	26255954.0600	62.9206
R109_50t_10w	11	1000	1493853596.4599	631133985.4200	136.6935
R110_100t_20w	69	1000	21767316347.5798	7622345982.8200	185.5723
R110_25t_5w	3	1000	38270827.5599	34488424.6600	10.9671
R110_50t_10w	13	1000	1298627388.6599	636761297.5800	103.9425
R111_100t_20w	76	1000	23372304372.9398	8473476077.2600	175.8290
R111_25t_5w	4	1000	42498384.0599	26144634.7600	62.5510
R111_50t_10w	16	1000	831122241.4399	496942686.1600	67.2471
R112_100t_20w	87	1000	20908080380.0198	10851236130.0600	92.6792
R112_25t_5w	4	1000	34543953.2400	26311467.9000	31.2885
R112_50t_10w	17	1000	1093877478.5599	427249792.1800	156.0276
R201_100t_20w	148	1000	253994839.7598	-1383419122.5000	118.3599
R201_25t_5w	11	1000	-554546.3800	-5171519.4600	89.2769
R201_50t_10w	40	1000	-124231486.3201	-125097475.6400	0.6922
R202_100t_20w	340	1000	4379948786.1398	-778169509.0800	662.8527
R202_25t_5w	29	1000	3895296.8999	-5171550.1800	175.3216
R202_50t_10w	77	1000	146748317.1199	-125097689.9000	217.3069
R203_100t_20w	536	1000	1907618989.9997	-891653653.5400	313.9417
R203_25t_5w	98	1000	-276413.5600	-5171614.5200	94.6551
R203_50t_10w	133	1000	77055479.7399	-125097805.5400	161.5961
R204_100t_20w	788	1000	2726324834.8998	-1007840112.0800	370.5116
R204_25t_5w	122	1000	8234027.0999	-5060450.1000	262.7133
R204_50t_10w	214	1000	78354311.8399	-125098025.1800	162.6343
R205_100t_20w	291	1000	1896810858.2997	-910567724.8000	308.3107
R205_25t_5w	24	1000	-721438.0400	-5171759.2000	86.0504
R205_50t_10w	69	1000	9094043.7598	-125097876.3800	107.2695
R206_100t_20w	459	1000	5193250386.1198	-964607487.1800	638.3796
R206_25t_5w	48	1000	3617164.3599	-5171813.9000	169.9399
R206_50t_10w	102	1000	77921377.8798	-125097837.4400	162.2883
R207_100t_20w	625	1000	3555838829.4997	-1018647805.1000	449.0744
R207_25t_5w	105	1000	-554479.6800	-5171794.9600	89.2787
R207_50t_10w	149	1000	76189580.1999	-125097967.7600	160.9039
R208_100t_20w	848	1000	1078104892.9997	-1042965700.1000	203.3691
R208_25t_5w	128	1000	8234013.5999	-5060560.5800	262.7095
R208_50t_10w	268	1000	78354044.4398	-125098091.9200	162.6340
R209_100t_20w	404	1000	4379948681.1399	-1013243421.1000	532.2701
R209_25t_5w	24	1000	8067195.4599	-5060483.0600	259.4155
R209_50t_10w	94	1000	13855538.1999	-125097999.8800	111.0757
R210_100t_20w	426	1000	4374544627.5798	-978117773.2200	547.2410
R210_25t_5w	50	1000	3895266.0399	-5171642.3200	175.3197
R210_50t_10w	101	1000	209514901.4999	-125097934.6800	267.4807
R211_100t_20w	506	1000	7676388592.9198	-942991911.9800	914.0460
R211_25t_5w	45	1000	7900317.8599	-5060502.8400	256.1172
R211_50t_10w	128	1000	143284906.6198	-122500836.0400	216.9664
RC101_100t_20w	31	1000	30092178382.2198	15814809991.8800	90.2784
RC101_25t_5w	4	1000	39438835.2199	27034405.3800	45.8838
RC101_50t_10w	6	1000	1501645656.2199	190034189.6200	690.1976
RC102_100t_20w	53	1000	24236944676.8597	15822916717.4800	53.1762
RC102_25t_5w	4	1000	38826968.9200	26645148.4800	45.7187
RC102_50t_10w	10	1000	1367454801.0999	578756222.9600	136.2747
RC103_100t_20w	73	1000	24201818776.8197	17527878844.9600	38.0761
RC103_25t_5w	5	1000	34599590.0599	21805915.8800	58.6706
RC103_50t_10w	14	1000	1030677953.1599	767922882.8400	34.2163
RC104_100t_20w	94	1000	20121798982.4197	18130424572.1200	10.9836
RC104_25t_5w	5	1000	34377119.2399	25810879.2400	33.1884
RC104_50t_10w	17	1000	1097773611.0800	498674346.2200	120.1383
RC105_100t_20w	47	1000	27519874515.9798	10907978098.4200	152.2912
RC105_25t_5w	4	1000	43165790.0599	30761388.4200	40.3245
RC105_50t_10w	9	1000	1441043372.7599	448893536.2400	221.0211
RC106_100t_20w	49	1000	25885164509.9798	11529438265.0600	124.5136
RC106_25t_5w	4	1000	39383323.5599	22417682.5200	75.6797
RC106_50t_10w	10	1000	1437147451.8799	515989390.5800	178.5226
RC107_100t_20w	67	1000	30767678749.7998	16638920438.6600	84.9139
RC107_25t_5w	4	1000	43388180.7199	30761589.8800	41.0466

Table E.7 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
RC107_50t_10w	13	1000	1164869108.6999	575726019.2200	102.3304
RC108_100t_20w	77	1000	21832164794.1197	14823176358.6800	47.2839
RC108_25t_5w	5	1000	38827096.1999	30372205.5600	27.8375
RC108_50t_10w	15	1000	1227636088.3399	630268451.2000	94.7798
RC201_100t_20w	143	1000	-1388819549.5002	-1378013035.7800	0.7781 *
RC201_25t_5w	13	1000	3784691.4399	-5171005.4000	173.1906
RC201_50t_10w	35	1000	-120766859.9402	-125096456.0400	3.4610
RC202_100t_20w	313	1000	4369142523.4196	-848419159.5200	614.9745
RC202_25t_5w	25	1000	3617839.9400	-5171480.4200	169.9575
RC202_50t_10w	69	1000	77921726.2198	-125096955.0600	162.2890
RC203_100t_20w	497	1000	1913023548.4997	-915968950.1800	308.8524
RC203_25t_5w	57	1000	3951142.0199	-5171505.3800	176.4021
RC203_50t_10w	118	1000	77055884.3598	-125096765.1800	161.5970
RC204_100t_20w	727	1000	1907619772.8397	-1032155821.9000	284.8189
RC204_25t_5w	99	1000	4062407.4399	-5060413.3600	180.2781
RC204_50t_10w	192	1000	76190308.5598	-125097014.9000	160.9049
RC205_100t_20w	256	1000	2739836285.9197	-856525543.8000	419.8779
RC205_25t_5w	24	1000	-53231.1200	-5171462.0600	98.9706
RC205_50t_10w	59	1000	77922090.5797	-125096766.4200	162.2894
RC206_100t_20w	287	1000	1078106682.4998	-905161565.1000	219.1065
RC206_25t_5w	22	1000	-386831.5600	-5171397.9400	92.5197
RC206_50t_10w	63	1000	-57999964.4802	-125097137.7600	53.6360
RC207_100t_20w	365	1000	6852279937.0799	-907864881.6600	854.7686
RC207_25t_5w	26	1000	7622588.0199	-5059890.2000	250.6473
RC207_50t_10w	86	1000	-119035962.7801	-122499526.2000	2.8274
RC208_100t_20w	457	1000	6857684162.9399	-978116228.2000	801.1113
RC208_25t_5w	43	1000	3450327.6799	-5060744.0000	168.1782
RC208_50t_10w	122	1000	148913526.1999	-125096819.1800	219.0386
test150-0-0-0-0_d0_tw0	3608	710	101050622678.1997	-28349336446.9000	456.4479
test150-0-0-0-0_d0_tw1	2090	1000	-24141770230.7996	-30832491493.7000	21.7002
test150-0-0-0-0_d0_tw2	1545	1000	7242547949.4989	-28832172176.8000	125.1196
test150-0-0-0-0_d0_tw3	1111	1000	6897666592.4002	-27383664735.0000	125.1889
test150-0-0-0-0_d0_tw4	1775	1000	163819258695.0000	-26142085667.0000	726.6495
test250-0-0-0-0_d0_tw0	-	-	-	23650649218.5000	-
test250-0-0-0-0_d0_tw1	-	-	-	-292254131472.0000	-
test250-0-0-0-0_d0_tw2	-	-	-	-275698693917.1000	-
test250-0-0-0-0_d0_tw3	3612	490	-214544933324.4147	-266914175565.0000	19.6202
test250-0-0-0-0_d0_tw4	-	-	-	-231776103927.7000	-
test50-0-0-0-0_d0_tw0	403	1000	-842598576.7999	-842599324.2000	0.0000
test50-0-0-0-0_d0_tw1	132	1000	-842598443.2999	-842599506.6000	0.0001
test50-0-0-0-0_d0_tw2	100	1000	-842598048.7999	-842599560.3000	0.0001
test50-0-0-0-0_d0_tw3	76	1000	-830114535.8999	-842598590.0000	1.4816
test50-0-0-0-0_d0_tw4	116	1000	-358882360.4999	-842599753.8000	57.4077

E.8 Tabu Search Results: Config. 8

Table E.8: Tabu Search experiments results with parameter configuration 8

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
10_District0	1282	50000	71064866922.7100	71563566909.3600	0.7017 *
10_District1	3600	21094	10391910011489.5490	8218857439409.0600	26.4398
10_District2	3600	15685	56242802309.1699	27771011394.0200	102.5234
10_District3	2319	50000	148016032311.1900	139155792160.4600	6.3671
10_District4	3600	4286	28824022749673.9530	21291228411410.1000	35.3798
10_District5	3600	8542	42657438984971.2600	37249297856014.8000	14.5187
11_District0	1220	50000	3327706843.2400	3046672128.3800	9.2243
11_District1	3600	13452	7519784104587.9960	6157994611789.7000	22.1141
11_District2	1980	50000	13457228190.3399	9214556192.4000	46.0431
11_District3	3600	40257	69047905553.9200	61007292651.7800	13.1797
11_District4	3601	5240	28327341544190.6520	22460659385239.4000	26.1198
11_District5	3600	8041	17354107747842.5300	15643005576898.3000	10.9384
12_District0	1081	50000	123249774847.7199	115036288619.9000	7.1399
12_District1	3600	6465	46524033514596.0800	37877356021322.0000	22.8280
12_District2	3600	41768	169586962232.2499	153054550706.1000	10.8016
12_District3	3600	25615	281503588989.2200	239836499336.7000	17.3731
12_District4	3600	1937	69976301190337.2660	59129219661289.8000	18.3447
12_District5	3600	4629	77604572919275.8100	65945373929847.0000	17.6800
13_District0	772	50000	163030094315.4800	154315121626.0000	5.6475
13_District1	3601	10059	19871772650002.9260	14674175609787.1000	35.4200
13_District2	3600	38519	154666155183.3599	126837068235.0100	21.9408
13_District3	3600	15791	545562728441.2301	429665639491.2800	26.9737
13_District4	3601	2500	34944914457256.2730	30033656135618.7000	16.3525
13_District5	3600	3707	56929657841132.4100	42764648834839.3000	33.1231
14_District0	707	50000	34998959934.3900	34977146773.4600	0.0623
14_District1	3600	8104	14524429777197.5840	12434388027267.4000	16.8085
14_District2	2769	50000	108121066238.6199	90165619840.7900	19.9138
14_District3	3600	34501	329736237547.9298	262406322982.9800	25.6586
14_District4	3600	4702	43274108721617.1600	31546738861338.4000	37.1745
14_District5	3600	5122	51310668743104.5400	44524141233501.8000	15.2423
15_District0	1134	50000	50596439088.6199	42188641727.2300	19.9290
15_District1	3600	7101	14069935349289.4470	12317798430422.9000	14.2244
15_District2	3600	25486	103156785657.2899	67421894826.9300	53.0019
15_District3	3600	22954	452400421567.4308	463823619193.1600	2.5250 *
15_District4	3600	5112	32194102455076.9380	22834329911998.4000	40.9899
15_District5	3600	6447	33961376146576.4570	28700034523627.8000	18.3321
16_District0	552	50000	116855140242.5500	120807470457.0500	3.3822 *
16_District1	3600	17250	16792011684544.6100	12316160134202.9000	36.3412

Table E.8 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
16_District2	2236	50000	103008971092.1900	94504646913.7700	8.9988
16_District3	3600	40255	273024373859.2299	210518442436.7200	29.6914
16_District4	3600	3540	35952100828187.3600	28327769996973.5000	26.9146
16_District5	3600	5252	54102712144284.7300	48522145053303.8000	11.5010
17_District0	417	50000	64480501874.8500	60633779564.4100	6.3441
17_District1	3600	13288	13583226327478.4280	12050832937058.4000	12.7160
17_District2	3600	43289	131790088641.2600	111787046906.6000	17.8938
17_District3	2002	50000	222276932231.4899	178730940805.7800	24.3639
17_District4	3602	3735	22294458031455.2340	19092974442252.1000	16.7678
17_District5	3600	9831	28598180584785.0620	23886759047749.6000	19.7239
18_District0	102	50000	2705606160.4100	2341527488.6000	15.5487
18_District1	3600	19535	15539109409529.4670	13924688690304.7000	11.5939
18_District2	1679	50000	5676720141.8700	4417358464.4100	28.5093
18_District3	944	50000	62677261168.5299	70920528499.0700	13.1519 *
18_District4	3600	5482	21449713637219.9300	18718722507656.5000	14.5896
18_District5	3600	16678	15543852583612.3090	12641617614178.6000	22.9577
19_District0	1121	50000	47699627935.7100	55462969817.4300	16.2754 *
19_District1	3600	8360	26413497392708.1800	22401449014876.9000	17.9097
19_District2	3600	32905	187674384382.1900	153280470262.9000	22.4385
19_District3	3600	34527	274814044921.7300	243569037679.2000	12.8279
19_District4	-	-	-	106192108824987.0000	-
19_District5	3601	4753	92374621349656.3100	78807991381031.8000	17.2147
1_District0	1895	50000	446621209022.5500	430590936121.1900	3.7228
1_District1	3600	4706	64113401266731.5800	56270206789859.1000	13.9384
1_District2	3600	33614	778146164132.2399	672001346398.5800	15.7953
1_District3	3600	22175	1274852363408.4600	1031045184732.7300	23.6466
1_District4	3600	965	89000469461106.8300	73732177837593.4000	20.7077
1_District5	3600	2982	105190510015786.4700	91806570717137.9000	14.5784
20_District0	497	50000	92787650758.6100	98421186256.3800	6.0714 *
20_District1	3600	10214	20166473748877.4140	17222315357479.6000	17.0950
20_District2	3600	26237	40158479189.1600	27378465012.0200	46.6790
20_District3	3600	19988	332285164824.6803	305213925793.8100	8.8695
20_District4	3600	3495	30556157284792.6700	24960379220572.1000	22.4186
20_District5	3600	4622	54096789514340.4500	48965990159239.8000	10.4782
21_District0	2311	50000	121961891548.5799	120414278382.3300	1.2852
21_District1	3600	9554	20356277402674.2970	16893868904739.4000	20.4950
21_District2	3600	34101	99104833899.7000	62666013838.2000	58.1476
21_District3	3600	33744	1083818307996.6499	892017226174.9800	21.5019
21_District4	3600	3538	27987492998716.2070	24304758516183.0000	15.1523
21_District5	3600	5584	57789774017343.4800	45930297567746.2000	25.8205
22_District0	545	50000	142334773151.6699	138906549645.3000	2.4680
22_District1	3600	14200	17751503267689.8480	14071948171020.4000	26.1481
22_District2	3308	50000	270352167268.5900	226837888943.8600	19.1829
22_District3	3600	24258	389586838914.7998	285892462254.7700	36.2704
22_District4	3600	4628	36430067637167.2660	29074924558505.3000	25.2972
22_District5	3600	5765	49609639676068.9840	42772519577575.2000	15.9848
23_District0	298	50000	34130461433.3000	32906581132.3800	3.7192
23_District1	3600	7709	20306570703942.8700	17211576588480.4000	17.9820
23_District2	2431	50000	229245872910.0900	183871311138.5000	24.6773
23_District3	3600	12893	286075456109.0404	250619333021.8100	14.1474
23_District4	3601	2316	44124735609381.3750	34555727151866.1000	27.6915
23_District5	3600	4121	82318311632011.2200	70608343796642.5000	16.5843
24_District0	1315	50000	116557323116.4800	105830236823.5000	10.1361
24_District1	3600	10321	14601551656495.2070	11389191027676.9000	28.2053
24_District2	2877	50000	25698213544.9499	26055867878.7800	1.3917 *
24_District3	3600	21504	176442969278.3299	159315579521.2200	10.7506
24_District4	3600	3353	28492637314463.2800	24517137738123.8000	16.2151
24_District5	3600	10128	29285060431772.2460	24738655441667.6000	18.3777
25_District0	279	50000	1465508222.8300	1255504790.1000	16.7266
25_District1	3600	13827	7110320772712.6380	6071232244353.8300	17.1149
25_District2	537	50000	2738854314.9000	2430507030.7800	12.6865
25_District3	1849	50000	53936038785.1399	54825213021.3900	1.6485 *
25_District4	3600	4865	25996861803541.5400	20503174454937.0000	26.7943
25_District5	3600	9498	12406452080578.2800	11011719621667.4000	12.6658
26_District0	2098	50000	229828525038.5700	225380467772.9800	1.9735
26_District1	3600	7156	48987132161122.1500	42180173025536.8000	16.1378
26_District2	2418	50000	388568810286.6699	282951685653.1000	37.3269
26_District3	3600	15712	1292287659214.3389	1171533328289.0400	10.3073
26_District4	3601	1413	126120081289807.2700	105580668249051.0000	19.4537
26_District5	3601	2140	88272702517243.4700	76363123397244.6000	15.5959
27_District0	1086	50000	150935841287.4100	162479691641.3000	7.6481 *
27_District1	3600	12021	24579099095816.6900	18425601260045.1000	33.3964
27_District2	3600	21545	85378522278.3000	61192189969.0700	39.5251
27_District3	3262	50000	135006682539.8000	116712931257.8400	15.6741
27_District4	3601	2431	42830700078064.5100	33778988955119.3000	26.7968
27_District5	3600	5601	46799178020401.0300	39568513214098.5000	18.2737
28_District0	655	50000	112954621923.5300	102108349093.8000	10.6223
28_District1	3600	9670	16751294190753.8380	15128335139531.3000	10.7279
28_District2	2518	50000	80008879999.6099	64692968024.3600	23.6747
28_District3	3600	23311	957802714175.6798	805624097687.0900	18.8895
28_District4	3600	4629	28571951497097.7540	22982825117023.6000	24.3187
28_District5	3600	7537	41410160521065.4840	33791534896023.1000	22.5459
29_District0	1094	50000	31995323111.8699	29353316624.4500	9.0007
29_District1	3600	18327	9019729753026.5980	7484826255380.2500	20.5068
29_District2	3600	39676	108362419182.7700	101233312043.5700	7.0422
29_District3	3600	20616	289675147038.2499	232601979076.2900	24.5368
29_District4	3600	5830	10869563929843.5680	8309110124637.3200	30.8150
29_District5	3600	4870	40098034357696.8600	35670467010142.6000	12.4124
2_District0	1890	50000	155041276217.5999	147005506881.3000	5.4663
2_District1	3600	6762	45459857429067.4450	39540340925649.8000	14.9708
2_District2	3600	43648	266311375917.0197	217829855689.6200	22.2566
2_District3	3600	36200	1210994966019.4000	858787147589.8100	41.0122
2_District4	3602	1581	74239030120333.2500	58876326066126.1000	26.0931
2_District5	3601	2824	113193541593170.9200	95903342380824.9000	18.0287
30_District0	362	50000	15544046438.0500	14856579555.0000	4.6273
30_District1	3600	16127	5453874315157.2070	4288991390631.0100	27.1598
30_District2	3600	46877	33258234580.2999	28984654012.0000	14.7442

Table E.8 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
30_District3	1953	50000	142168951382.7100	140260859804.8700	1.3603
30_District4	3600	8423	6957178114976.4290	6033193692111.0700	15.3150
30_District5	3600	16373	6652589252190.8170	5491783393822.4800	21.1371
3_District0	606	50000	97411176585.6400	90701681116.3600	7.3973
3_District1	3600	14718	21669256497445.5620	17931112587450.7000	20.8472
3_District2	3600	31047	218831386416.8799	184423936337.4900	18.6567
3_District3	3600	21472	1055208515491.4495	837894240881.4400	25.9357
3_District4	-	-	-	67042838268903.4000	-
3_District5	3600	4937	71450369212445.9400	60566955554266.6000	17.9692
4_District0	355	50000	21143876404.2400	20379141775.5100	3.7525
4_District1	3600	9057	18643409643794.8630	16091560843817.3000	15.8583
4_District2	1807	50000	22470399290.9999	19771989271.8700	13.6476
4_District3	2626	50000	31168654413.9700	31243041764.1500	0.2386 *
4_District4	3601	1614	55827883212635.5600	44497725128783.0000	25.4623
4_District5	3600	9069	35485732192312.5100	30292083288905.3000	17.1452
5_District0	1148	50000	102454919215.7400	105305055950.1000	2.7818 *
5_District1	3600	9333	26958729717452.4450	22578812619408.6000	19.3983
5_District2	3600	43657	72814722089.2699	49746631100.9200	46.3711
5_District3	3600	24218	368339564893.9703	366270241695.4900	0.5649
5_District4	3600	1797	87052360312381.4500	68243049339872.1000	27.5622
5_District5	3600	2429	144263825380312.4000	121803986176569.0000	18.4393
6_District0	846	50000	184831595148.0799	174024536017.0000	6.2100
6_District1	3600	5584	29757855339023.4100	24635464198083.6000	20.7927
6_District2	3600	25764	314699930704.3200	263920214770.7000	19.2405
6_District3	3600	39654	326846711901.7501	272026860685.1900	20.1523
6_District4	3600	3160	41017633874659.3900	32671893526336.1000	25.5440
6_District5	3600	5613	65047160187885.1250	52577668995018.6000	23.7163
7_District0	1556	50000	50362920704.4700	53992400830.1800	7.2066 *
7_District1	3601	5236	28469891691201.0230	25530776489059.8000	11.5120
7_District2	2690	50000	95085164959.8600	96812988604.0900	1.8171 *
7_District3	3600	42807	885810255449.4302	627921570619.9200	41.0702
7_District4	3601	3415	38983888110125.7900	30888008083772.9000	26.2104
7_District5	3600	3801	59599457455793.6500	52214576657422.1000	14.1433
8_District0	595	50000	55023099709.3399	47815962594.9100	15.0726
8_District1	3600	12075	15922201158605.7680	13654929607346.6000	16.6040
8_District2	2756	50000	66765823275.3500	73150043166.7900	9.5621 *
8_District3	3600	31339	482102510681.2601	367102727288.8900	31.3263
8_District4	3600	3968	35399491028950.2900	29816748603227.9000	18.7235
8_District5	3600	9055	50474401568700.1000	40450402089088.4000	24.7809
9_District0	312	50000	102743034998.7600	100093204504.3000	2.6473
9_District1	3600	13907	15318565989602.6800	12178217016824.2000	25.7866
9_District2	1872	50000	106982243005.6299	89342594098.5600	19.7438
9_District3	3600	32485	1284512871128.3906	911115793606.4100	40.9823
9_District4	3602	2227	48344285222867.3100	37164202673658.1000	30.0829
9_District5	3600	8580	32702225945411.4060	25382936280280.4000	28.8354
C101_100t_20w	3283	50000	99095856585.7203	-12304901708.1200	905.3364
C101_25t_5w	175	50000	113642735.5600	9512566.6400	1094.6590
C101_50t_10w	688	50000	3214099348.6999	-1079154704.1200	397.8349
C102_100t_20w	3600	11668	98901313024.3402	-5155408825.8200	2018.3990
C102_25t_5w	622	50000	39550192.4200	-37546115.7600	205.3376
C102_50t_10w	3600	47988	3140078105.8398	-1102530180.6200	384.8065
C103_100t_20w	3600	5585	39930163004.1205	-7222438475.8600	652.8626
C103_25t_5w	2222	50000	40551560.5799	-43052838.4600	194.1902
C103_50t_10w	3600	12944	3743938477.6599	-1110321792.2400	437.1940
C104_100t_20w	3600	3724	17314423586.3605	-8681519205.2400	299.4400
C104_25t_5w	3271	50000	39550261.0599	-33540706.8600	217.9171
C104_50t_10w	3600	7465	1305120952.8799	-1086946666.3200	220.0722
C105_100t_20w	3600	29483	77185339028.9999	-12158993843.3000	734.8003
C105_25t_5w	238	50000	116646388.4000	1002190.7200	11539.1407
C105_50t_10w	1541	50000	3186828351.9399	-1102529614.6600	389.0469
C106_100t_20w	3600	21093	54739825368.8202	-5349954009.7600	1123.1831
C106_25t_5w	250	50000	115144572.7800	9512566.6400	1110.4469
C106_50t_10w	1230	50000	1379142584.3799	-1086946589.7000	226.8822
C107_100t_20w	3600	19782	76747614835.6599	-6979258964.6200	1199.6527
C107_25t_5w	376	50000	112641450.9999	-2001420.3200	5728.0757
C107_50t_10w	2493	50000	3747834654.6599	-1125905161.9000	432.8730
C108_100t_20w	3600	13656	47079655318.6598	-6492899831.1000	825.0944
C108_25t_5w	593	50000	71089677.5200	-8009072.9800	987.6143
C108_50t_10w	3532	50000	2551800875.0799	-1125905177.3200	326.6443
C109_100t_20w	3600	8826	47006701457.3999	-6614488230.8200	810.6627
C109_25t_5w	859	50000	35044676.3999	-26031566.9600	234.6237
C109_50t_10w	3600	31838	2524529654.7199	-1125905761.1800	324.2221
C201_100t_20w	3600	14493	-12596716053.9600	-12596720162.2300	0.0000
C201_25t_5w	1028	50000	-45555916.6400	-45555916.9200	0.0000
C201_50t_10w	2614	50000	-506460875.0600	-1125905241.3800	55.0174
C202_100t_20w	3602	2358	39322214142.4203	-6055175633.1600	749.3984
C202_25t_5w	3600	27539	-45555913.6599	-45555950.8800	0.0000
C202_50t_10w	3601	6934	685677219.4399	-1125905361.9600	160.9000
C203_100t_20w	3602	1061	24536869879.3203	-6930623788.8800	454.0355
C203_25t_5w	3600	4203	-45555827.1999	-45555956.2400	0.0002
C203_50t_10w	3600	2502	709052735.5999	-1125905456.0600	162.9762
C204_100t_20w	3602	519	24609823628.3203	-8657198950.2400	384.2700
C204_25t_5w	3600	3434	-42551953.2800	-45555977.0000	6.5941
C204_50t_10w	3601	1321	93504394.2998	-1125905585.1400	108.3048
C205_100t_20w	3600	5597	9702889586.1598	-12596718687.8400	177.0271
C205_25t_5w	2915	50000	-45555910.0000	-45555926.8000	0.0000
C205_50t_10w	3600	29566	-522044220.5000	-1125905403.9800	53.6333
C206_100t_20w	3600	3356	31953859516.5598	-7733117210.9600	513.2080
C206_25t_5w	3600	43142	-45555918.6800	-45555926.8000	0.0000
C206_50t_10w	3600	17394	677885628.8399	-1125905541.4200	160.2080
C207_100t_20w	3600	2680	24488233802.3396	-7222438653.2400	439.0576
C207_25t_5w	3600	33317	-45555880.1400	-45555933.4200	0.0001
C207_50t_10w	3600	5739	-498668527.7600	-1125905455.9000	55.7095
C208_100t_20w	3600	2362	24536869467.2000	-8657201177.3800	383.4272
C208_25t_5w	3600	28572	-45555911.2800	-45555933.1600	0.0000
C208_50t_10w	3600	10337	697364998.0999	-1125905465.6600	161.9381
hh_00_P0	3600	23320	855649513467.0670	6663651008.2900	12740.5511

Table E.8 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
ll1_00_P0	2018	50000	12029742925.6600	1338732826.5000	798.5917
ll1_01_P0	1973	50000	8761696992.0399	1338732826.5000	554.4768
ll1_02_P0	2058	50000	8761710941.1901	1338732826.5000	554.4779
ll1_03_P0	2034	50000	10364620911.9201	1338732826.5000	674.2113
ll1_04_P0	2059	50000	10333442162.1100	1338732826.5000	671.8823
ll1_05_P0	2067	50000	15360038245.1100	1307660206.1200	1074.6199
ll1_06_P0	1977	50000	12029721461.7801	1214281948.4400	890.6860
ll1_07_P0	2063	50000	5431393662.0100	1338732826.5000	305.7115
ll2_00_P0	1456	50000	402249701.6198	85352916.2700	371.2782
ll3_00_P0	1422	50000	965681078.0999	-198615845.6200	586.2054
R101_100t_20w	646	50000	28343983676.1399	6319981502.8600	348.4820
R101_25t_5w	24	50000	48450070.2600	47893886.5800	1.1612
R101_50t_10w	99	50000	1176556007.9999	709051142.3992	65.9338
R102_100t_20w	1894	50000	20929696190.8198	14101742003.1800	48.4192
R102_25t_5w	46	50000	38715838.3600	34655177.0000	11.7173
R102_50t_10w	248	50000	1037170523.4999	641955625.9388	61.5642
R103_100t_20w	3299	50000	20094778348.1398	12469734030.7400	61.1484
R103_25t_5w	101	50000	30705785.2999	30260803.3400	1.4704
R103_50t_10w	585	50000	963149100.1199	570531480.7600	68.8161
R104_100t_20w	3600	39613	19294986670.3198	14023383931.9600	37.5915
R104_25t_5w	115	50000	26033244.6400	29982663.2800	15.1706 *
R104_50t_10w	1036	50000	696931044.8200	293491458.8800	137.4621
R105_100t_20w	1182	50000	24201817987.1599	8424839701.0800	187.2674
R105_25t_5w	36	50000	40106282.9600	35155829.6400	14.0814
R105_50t_10w	168	50000	1098638838.5000	388290958.0600	182.9421
R106_100t_20w	2193	50000	24163990277.4198	8443754034.7200	186.1759
R106_25t_5w	51	50000	30594658.6600	30483330.4600	0.3652
R106_50t_10w	307	50000	1031976025.0999	303880567.4400	239.5992
R107_100t_20w	3266	50000	20113692439.8597	11634816307.3400	72.8750
R107_25t_5w	92	50000	26144590.7199	22028382.6800	18.6859
R107_50t_10w	570	50000	787834787.2399	372274723.4800	111.6272
R108_100t_20w	3600	41055	17654872808.1198	12369760088.0800	42.7260
R108_25t_5w	108	50000	26144515.0399	26033406.5200	0.4267
R108_50t_10w	989	50000	765758147.6999	293491505.9600	160.9132
R109_100t_20w	2222	50000	22548193919.5199	6636116061.1200	239.7799
R109_25t_5w	54	50000	34265826.9200	26255954.0600	30.5068
R109_50t_10w	291	50000	900381997.2799	631133985.4200	42.6609
R110_100t_20w	2694	50000	21767316347.5798	7622345982.8200	185.5723
R110_25t_5w	51	50000	34210247.1599	34488424.6600	0.8131 *
R110_50t_10w	357	50000	900382247.5399	636761297.5800	41.4002
R111_100t_20w	3009	50000	23372304372.9398	8473476077.2600	175.8290
R111_25t_5w	68	50000	29815907.5400	26144634.7600	14.0421
R111_50t_10w	498	50000	831122241.4399	496942686.1600	67.2471
R112_100t_20w	3590	50000	20908080380.0198	10851236130.0600	92.6792
R112_25t_5w	67	50000	29927017.5799	26311467.9000	13.7413
R112_50t_10w	452	50000	902979348.9400	427249792.1800	111.3469
R201_100t_20w	3600	23601	253994839.7598	-1383419122.5000	118.3599
R201_25t_5w	492	50000	-5171446.6000	-5171519.4600	0.0014
R201_50t_10w	1743	50000	-124231486.3201	-125097475.6400	0.6922
R202_100t_20w	3600	8941	4379948786.1398	-778169509.0800	662.8527
R202_25t_5w	999	50000	-5171436.6400	-5171550.1800	0.0021
R202_50t_10w	3600	40874	146748317.1199	-125097689.9000	217.3069
R203_100t_20w	3600	4740	1907618989.9997	-891653653.5400	313.9417
R203_25t_5w	2393	50000	-4893127.5399	-5171614.5200	5.3849
R203_50t_10w	3600	17050	77055479.7399	-125097805.5400	161.5961
R204_100t_20w	3600	3712	2726324834.8998	-1007840112.0800	370.5116
R204_25t_5w	3182	50000	-5060010.4000	-5060450.1000	0.0086
R204_50t_10w	3600	9645	78354311.8399	-125098025.1800	162.6343
R205_100t_20w	3600	10834	1896810858.2997	-910567724.8000	308.3107
R205_25t_5w	1064	50000	-5171482.1600	-5171759.2000	0.0053
R205_50t_10w	3600	43164	9094043.7598	-125097876.3800	107.2695
R206_100t_20w	3600	6674	5193250386.1198	-964607487.1800	638.3796
R206_25t_5w	1709	50000	-5171543.0199	-5171813.9000	0.0052
R206_50t_10w	3600	26469	77921377.8798	-125097837.4400	162.2883
R207_100t_20w	3600	4532	3555838829.4997	-1018647805.1000	449.0744
R207_25t_5w	2565	50000	-5171737.6399	-5171794.9600	0.0011
R207_50t_10w	3600	14162	76189580.1999	-125097967.7600	160.9039
R208_100t_20w	3600	3588	1078104892.9997	-1042965700.1000	203.3691
R208_25t_5w	2969	50000	-5060332.4600	-5060560.5800	0.0045
R208_50t_10w	3600	8123	78354044.4398	-125098091.9200	162.6340
R209_100t_20w	3600	8063	4379948681.1399	-1013243421.1000	532.2701
R209_25t_5w	1105	50000	-4448269.2800	-5060483.0600	12.0979
R209_50t_10w	3600	29493	13855538.1999	-125097999.8800	111.0757
R210_100t_20w	3600	7086	4374544627.5798	-978117773.2200	547.2410
R210_25t_5w	1741	50000	-5171454.9399	-5171642.3200	0.0036
R210_50t_10w	3600	25005	209514918.5799	-125097934.6800	267.4807
R211_100t_20w	3600	6457	7676388592.9198	-942991911.9800	914.0460
R211_25t_5w	1745	50000	-5060306.1199	-5060502.8400	0.0038
R211_50t_10w	3600	20108	143284906.6198	-122500836.0400	216.9664
RC101_100t_20w	1037	50000	30092178382.2198	15814809991.8800	90.2784
RC101_25t_5w	34	50000	30650072.4200	27034405.3800	13.3743
RC101_50t_10w	114	50000	970074742.1999	190034189.6200	410.4737
RC102_100t_20w	1767	50000	24236944676.8597	15822916717.4800	53.1762
RC102_25t_5w	42	50000	26533966.6800	26645148.4800	0.4190 *
RC102_50t_10w	143	50000	977001259.5399	578756222.9600	68.8104
RC103_100t_20w	2721	50000	24201818776.8197	17527878844.9600	38.0761
RC103_25t_5w	60	50000	21917092.7600	21805915.8800	0.5098
RC103_50t_10w	258	50000	838914045.9000	767922882.8400	9.2445
RC104_100t_20w	3600	48983	20121798982.4197	18130424572.1200	10.9836
RC104_25t_5w	74	50000	22195218.5200	25810879.2400	16.2902 *
RC104_50t_10w	412	50000	836749757.1400	498674346.2200	67.7948
RC105_100t_20w	1684	50000	27519874515.9798	10907978098.4200	152.2912
RC105_25t_5w	38	50000	26645206.1800	30761388.4200	15.4481 *
RC105_50t_10w	161	50000	844108351.2399	448893536.2400	88.0419
RC106_100t_20w	1843	50000	26709274569.8198	11529438265.0600	131.6615
RC106_25t_5w	50	50000	26589569.3399	22417682.5200	18.6098
RC106_50t_10w	170	50000	847571382.6200	515989390.5800	64.2613

Table E.8 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
RC107_100t_20w	2403	50000	30767678749.7998	16638920438.6600	84.9139
RC107_25t_5w	46	50000	26255800.9200	30761589.8800	17.1611 *
RC107_50t_10w	239	50000	899516382.3000	575726019.2200	56.2403
RC108_100t_20w	3064	50000	21832164794.1197	14823176358.6800	47.2839
RC108_25t_5w	63	50000	25977760.8799	30372205.5600	16.9161 *
RC108_50t_10w	313	50000	840645214.8799	630268451.2000	33.3789
RC201_100t_20w	3600	24559	-1388819549.5002	-1378013035.7800	0.7781 *
RC201_25t_5w	465	50000	-5170833.5800	-5171005.4000	0.0033
RC201_50t_10w	1425	50000	-120766859.9402	-125096456.0400	3.4610
RC202_100t_20w	3600	10081	4369142523.4196	-848419159.5200	614.9745
RC202_25t_5w	702	50000	-5171301.2800	-5171480.4200	0.0034
RC202_50t_10w	3188	50000	77921726.2198	-125096955.0600	162.2890
RC203_100t_20w	3600	5581	1913023548.4997	-915968950.1800	308.8524
RC203_25t_5w	1701	50000	-5170950.3199	-5171505.3800	0.0107
RC203_50t_10w	3600	23238	77055884.3598	-125096765.1800	161.5970
RC204_100t_20w	3600	4305	1907619884.9597	-1032155821.9000	284.8189
RC204_25t_5w	2589	50000	-5060060.2399	-5060413.3600	0.0069
RC204_50t_10w	3600	11359	76190308.5598	-125097014.9000	160.9049
RC205_100t_20w	3600	12931	2739836285.9197	-856525543.8000	419.8779
RC205_25t_5w	599	50000	-5004006.2000	-5171462.0600	3.2380
RC205_50t_10w	2961	50000	77922090.5797	-125096766.4200	162.2894
RC206_100t_20w	3600	11622	1078106682.4998	-905161565.1000	219.1065
RC206_25t_5w	856	50000	-5171348.0800	-5171397.9400	0.0009
RC206_50t_10w	3249	50000	-57999964.4802	-125097137.7600	53.6360
RC207_100t_20w	3600	9470	6852279937.0799	-907864881.6600	854.7686
RC207_25t_5w	864	50000	-4781707.9399	-5059890.2000	5.4977
RC207_50t_10w	3600	36333	-119035962.7801	-122499526.2000	2.8274
RC208_100t_20w	3600	6966	6857684162.9399	-978116228.2000	801.1113
RC208_25t_5w	1269	50000	-5060219.5799	-5060744.0000	0.0103
RC208_50t_10w	3600	24996	148913526.1999	-125096819.1800	219.0386
test150-0-0-0-0_d0_tw0	3601	1209	101050622678.1997	-28349336446.9000	456.4479
test150-0-0-0-0_d0_tw1	3600	2975	-24141770230.7996	-30832491493.7000	21.7002
test150-0-0-0-0_d0_tw2	3600	3993	7242548454.7989	-28832172176.8000	125.1196
test150-0-0-0-0_d0_tw3	3600	5397	6897666592.4002	-27383664735.0000	125.1889
test150-0-0-0-0_d0_tw4	3600	3496	163819258695.0000	-26142085667.0000	726.6495
test250-0-0-0-0_d0_tw0	-	-	-	23650649218.5000	-
test250-0-0-0-0_d0_tw1	-	-	-	-292254131472.0000	-
test250-0-0-0-0_d0_tw2	-	-	-	-275698693917.1000	-
test250-0-0-0-0_d0_tw3	3600	871	-214544933324.4147	-266914175565.0000	19.6202
test250-0-0-0-0_d0_tw4	-	-	-	-231776103927.7000	-
test50-0-0-0-0_d0_tw0	3600	7002	-842598576.7999	-842599324.2000	0.0000
test50-0-0-0-0_d0_tw1	3600	26722	-842598443.2999	-842599506.6000	0.0001
test50-0-0-0-0_d0_tw2	3600	36788	-842598048.7999	-842599560.3000	0.0001
test50-0-0-0-0_d0_tw3	3519	50000	-830114535.8999	-842598590.0000	1.4816
test50-0-0-0-0_d0_tw4	3600	34601	-358882360.4999	-842599753.8000	57.4077

E.9 Tabu Search Results: Config. 9

Table E.9: Tabu Search experiments results with parameter configuration 9

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
10_District0	66	10000	74618110144.1400	71563566909.3600	4.2682
10_District1	581	10000	8938146360546.6970	8218857439409.0600	8.7516
10_District2	319	10000	46451869602.0199	27771011394.0200	67.2674
10_District3	165	10000	141620940002.7101	139155792160.4600	1.7715
10_District4	3132	10000	26625521931713.8120	21291228411410.1000	25.0539
10_District5	2507	10000	41737394243010.0900	37249297856014.8000	12.0488
11_District0	68	10000	3928098965.6099	3046672128.3800	28.9308
11_District1	874	10000	6644833546027.0110	6157994611789.7000	7.9058
11_District2	116	10000	14972468061.3300	9214556192.4000	62.4871
11_District3	163	10000	65361664595.6499	61007292651.7800	7.1374
11_District4	3044	10000	25977912326971.8750	22460659385239.4000	15.6596
11_District5	1813	10000	16899788327820.8890	15643005576898.3000	8.0341
12_District0	75	10000	131463261688.0899	115036288619.9000	14.2798
12_District1	2249	10000	43251322721596.4900	37877356021322.0000	14.1878
12_District2	167	10000	197276898620.9196	153054550706.1000	28.8931
12_District3	417	10000	193884174241.5399	239836499336.7000	23.7009 *
12_District4	3600	6905	67691720648819.9450	59129219661289.8000	14.4809
12_District5	3600	9443	73828178354991.5800	65945373929847.0000	11.9535
13_District0	30	10000	176653499888.4400	154315121626.0000	14.4758
13_District1	1411	10000	18424852168090.4020	14674175609787.1000	25.5597
13_District2	373	10000	128663035703.5597	126837068235.0100	1.4396
13_District3	1009	10000	450725719458.7006	429665639491.2800	4.9015
13_District4	1244	4003	34087365126496.8300	30033656135618.7000	13.4972
13_District5	2418	10000	46944882845825.2100	42764648834839.3000	9.7749
14_District0	47	10000	36438617953.3200	34977146773.4600	4.1783
14_District1	1955	10000	13518736316108.9470	12434388027267.4000	8.7205
14_District2	97	10000	108031438247.2699	90165619840.7900	19.8144
14_District3	388	10000	265662606247.0595	262406322982.9800	1.2409
14_District4	3329	10000	39936859284178.1900	31546738861338.4000	26.5958
14_District5	3212	10000	48558384707731.8600	44524141233501.8000	9.0608
15_District0	42	10000	58881792510.4799	42188641727.2300	39.5678
15_District1	1071	10000	13162303174786.7270	12317798430422.9000	6.8559
15_District2	208	10000	80880489793.5000	67421894826.9300	19.9617
15_District3	561	10000	436031815251.0704	463823619193.1600	6.3738 *
15_District4	3015	10000	29797811910303.2500	22834329911998.4000	30.4956
15_District5	2111	10000	32392334887695.6200	28700034523627.8000	12.8651

Table E.9 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
16_District0	34	10000	126871892522.6999	120807470457.0500	5.0199
16_District1	1033	10000	14794477569995.2770	12316160134202.9000	20.1224
16_District2	96	10000	111048429008.9600	94504646913.7700	17.5057
16_District3	377	10000	253343907173.6297	210518442436.7200	20.3428
16_District4	3484	10000	35094026384039.7300	28327769996973.5000	23.8855
16_District5	3148	10000	52349926885974.4300	48522145053303.8000	7.8887
17_District0	28	10000	64731054335.9600	60633779564.4100	6.7574
17_District1	1070	10000	12688141175700.4320	12050832937058.4000	5.2884
17_District2	230	10000	121137939214.3299	111787046906.6000	8.3649
17_District3	126	10000	214101128623.6199	178730940805.7800	19.7896
17_District4	3601	9086	21283194597803.1400	19092974442252.1000	11.4713
17_District5	1452	10000	27638117328733.9140	23886759047749.6000	15.7047
18_District0	11	10000	2879317002.3000	2341527488.6000	22.9674
18_District1	778	10000	14403311116798.6740	13924688690304.7000	3.4372
18_District2	92	10000	6980830817.8900	4417358464.4100	58.0317
18_District3	92	10000	65353954501.7099	70920528499.0700	8.5175 *
18_District4	2552	10000	19145521075293.1200	18718722507656.5000	2.2800
18_District5	1003	10000	14949089159976.1300	12641617614178.6000	18.2529
19_District0	51	10000	52251568626.6599	55462969817.4300	6.1460 *
19_District1	1807	10000	25223614895354.1330	22401449014876.9000	12.5981
19_District2	200	10000	165814478917.1300	153280470262.9000	8.1771
19_District3	392	10000	238376329639.1599	243569037679.2000	2.1783 *
19_District4	3603	3866	126212233021404.3400	106192108824987.0000	18.8527
19_District5	2721	10000	88022057535817.6200	78807991381031.8000	11.6917
1_District0	86	10000	435637504588.2101	430590936121.1900	1.1720
1_District1	2979	10000	60290879750585.8750	56270206789859.1000	7.1452
1_District2	193	10000	760206194727.2006	672001346398.5800	13.1256
1_District3	731	10000	1128482019371.4897	1031045184732.7300	9.4502
1_District4	3604	3858	86040617164160.9500	73732177837593.4000	16.6934
1_District5	3600	8727	103173937748597.5800	91806570717137.9000	12.3818
20_District0	25	10000	93014387889.8900	98421186256.3800	5.8128 *
20_District1	1193	10000	19226885614534.7930	17222315357479.6000	11.6393
20_District2	326	10000	39869569648.9399	27378465012.0200	45.6238
20_District3	1093	10000	285676910265.4900	305213925793.8100	6.8388 *
20_District4	3600	9527	28884777935266.1400	24960379220572.1000	15.7225
20_District5	3389	10000	51685009954196.1300	48965990159239.8000	5.5528
21_District0	145	10000	113250328255.1900	120414278382.3300	6.3257 *
21_District1	1234	10000	19157972009032.0900	16893868904739.4000	13.4019
21_District2	303	10000	94341231938.1700	62666013838.2000	50.5460
21_District3	496	10000	977964334030.1404	892017226174.9800	9.6351
21_District4	3601	9121	26068378245609.3800	24304758516183.0000	7.2562
21_District5	3124	10000	54873564648639.2800	45930297567746.2000	19.4713
22_District0	33	10000	150892635308.1400	138906549645.3000	8.6288
22_District1	1037	10000	16851758043880.0270	14071948171020.4000	19.7542
22_District2	122	10000	299154703212.1001	226837888943.8600	31.8803
22_District3	659	10000	352560888776.8301	285892462254.7700	23.3194
22_District4	3266	10000	34336355866864.7540	29074924558505.3000	18.0961
22_District5	2825	10000	46531232287241.9600	42772519577575.2000	8.7876
23_District0	31	10000	34239031251.4899	32906581132.3800	4.0491
23_District1	1756	10000	19302724839878.5300	17211576588480.4000	12.1496
23_District2	101	10000	229041299021.5000	183871311138.5000	24.5660
23_District3	547	10000	234426382400.4702	250619333021.8100	6.9074 *
23_District4	3606	8674	41527949163807.1200	34555727151866.1000	20.1767
23_District5	3225	10000	80689878412172.8100	70608343796642.5000	14.2781
24_District0	64	10000	124270185761.0799	105830236823.5000	17.4240
24_District1	998	10000	14050104442426.8890	11389191027676.9000	23.3634
24_District2	184	10000	23896688892.0099	26055867878.7800	9.0354 *
24_District3	589	10000	153953627803.2999	159315579521.2200	3.4828 *
24_District4	3600	8171	27230600919334.0160	24517137738123.8000	11.0676
24_District5	1518	10000	28263587386481.5860	24738655441667.6000	14.2486
25_District0	21	10000	1806481296.6499	1255504790.1000	43.8848
25_District1	895	10000	6895336939285.3080	6071232244353.8300	13.5739
25_District2	34	10000	3359177843.2700	2430507030.7800	38.2089
25_District3	111	10000	59444584410.6199	54825213021.3900	8.4256
25_District4	3188	10000	23989775031108.7970	20503174454937.0000	17.0051
25_District5	976	10000	11427181237478.5660	11011719621667.4000	3.7729
26_District0	127	10000	223101299371.0099	225380467772.9800	1.0215 *
26_District1	2289	10000	48987132161122.1500	42180173025536.8000	16.1378
26_District2	158	10000	352738484054.6600	282951685653.1000	24.6638
26_District3	1138	10000	1221285639579.8296	1171533328289.0400	4.2467
26_District4	3604	5569	117912475464892.0200	105580668249051.0000	11.6799
26_District5	3600	8474	83861126026278.4400	76363123397244.6000	9.8188
27_District0	107	10000	166866354604.0698	162479691641.3000	2.6998
27_District1	986	10000	21716688296602.4800	18425601260045.1000	17.8614
27_District2	241	10000	76764212397.3599	61192189969.0700	25.4477
27_District3	380	10000	112350112523.6599	116712931257.8400	3.8832 *
27_District4	3602	8001	41812407340678.4100	33778988955119.3000	23.7822
27_District5	2444	10000	42816916358355.2340	39568513214098.5000	8.2095
28_District0	25	10000	124151481425.7999	102108349093.8000	21.5879
28_District1	1530	10000	15643075617476.7600	15128335139531.3000	3.4024
28_District2	189	10000	80344076737.8700	64692968024.3600	24.1929
28_District3	707	10000	837122182158.8696	805624097687.0900	3.9097
28_District4	3238	10000	27028833734036.9020	22982825117023.6000	17.6044
28_District5	2065	10000	39235471171606.7000	33791534896023.1000	16.1103
29_District0	109	10000	34855763286.8800	29353316624.4500	18.7455
29_District1	888	10000	7784786607084.8380	7484826255380.2500	4.0075
29_District2	284	10000	83239449171.6500	101233312043.5700	21.6169 *
29_District3	478	10000	269030594406.3901	232601979076.2900	15.6613
29_District4	2183	10000	10036792921737.5100	8309110124637.3200	20.7926
29_District5	3525	10000	38683646564373.4900	35670467010142.6000	8.4472
2_District0	86	10000	155163959743.5298	147005506881.3000	5.5497
2_District1	2321	10000	43576984052478.6200	39540340925649.8000	10.2089
2_District2	331	10000	203945898858.5395	217829855689.6200	6.8076 *
2_District3	300	10000	998262259698.0299	858787147589.8100	16.2409
2_District4	3607	5958	70460215578833.3900	58876326066126.1000	19.6749
2_District5	3600	8969	111088965128385.5800	95903342380824.9000	15.8342
30_District0	23	10000	16310602209.7699	14856579555.0000	9.7870

Table E.9 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
30_District1	441	10000	4897299318915.8900	4288991390631.0100	14.1830
30_District2	103	10000	40086500784.8400	28984654012.0000	38.3024
30_District3	252	10000	130364660141.9401	140260859804.8700	7.5911 *
30_District4	1590	10000	6216507889135.2390	6033193692111.0700	3.0384
30_District5	862	10000	5947167169945.3180	5491783393822.4800	8.2920
3_District0	38	10000	100786506020.6299	90701681116.3600	11.1186
3_District1	992	10000	19490649149152.9180	17931112587450.7000	8.6973
3_District2	241	10000	195827548126.4300	184423936337.4900	6.1833
3_District3	662	10000	874720309591.7599	837894240881.4400	4.3950
3_District4	3600	3863	80535804587225.3900	67042838268903.4000	20.1258
3_District5	3195	10000	67935601558899.2500	60566955554266.6000	12.1661
4_District0	14	10000	23630985835.7400	20379141775.5100	15.9567
4_District1	1184	10000	17327163824302.3030	16091560843817.3000	7.6785
4_District2	112	10000	20949476998.0799	19771989271.8700	5.9553
4_District3	174	10000	33281278576.5800	31243041764.1500	6.5238
4_District4	3600	7292	53210299917913.2500	44497725128783.0000	19.5798
4_District5	1589	10000	34717610007935.9020	30292083288905.3000	14.6095
5_District0	116	10000	108530210933.7999	105305055950.1000	3.0626
5_District1	1652	10000	26647092666473.4770	22578812619408.6000	18.0181
5_District2	211	10000	76449400420.9900	49746631100.9200	53.6775
5_District3	579	10000	353728887601.8504	366270241695.4900	3.5454 *
5_District4	3600	7023	81372702828727.6600	68243049339872.1000	19.2395
5_District5	3601	8160	138140972195760.0000	121803986176569.0000	13.4125
6_District0	53	10000	201289203063.1800	174024536017.0000	15.6671
6_District1	2890	10000	27976253424033.1050	24635464198083.6000	13.5608
6_District2	341	10000	272598742186.6601	263920214770.7000	3.2883
6_District3	549	10000	306616163602.7795	272026860685.1900	12.7153
6_District4	3600	9805	38143579430174.5160	32671893526336.1000	16.7473
6_District5	2953	10000	63128223388004.1100	52577668995018.6000	20.0666
7_District0	107	10000	48160415535.0799	53992400830.1800	12.1094 *
7_District1	2416	10000	27420153708106.0040	25530776489059.8000	7.4003
7_District2	92	10000	112994195720.6300	96812988604.0900	16.7138
7_District3	339	10000	735339383977.3596	627921570619.9200	17.1068
7_District4	3367	10000	34572564912594.2150	30888008083772.9000	11.9287
7_District5	3607	9158	58330881735633.0300	52214576657422.1000	11.7137
8_District0	28	10000	59838631063.5099	47815962594.9100	25.1436
8_District1	1022	10000	15133584966635.4690	13654929607346.6000	10.8287
8_District2	189	10000	75954328001.1000	73150043166.7900	3.8336
8_District3	463	10000	384536370514.6995	367102727288.8900	4.7489
8_District4	3448	10000	32888051961838.9770	29816748603227.9000	10.3005
8_District5	1843	10000	47037083949305.2600	40450402089088.4000	16.2833
9_District0	20	10000	105827602912.7499	100093204504.3000	5.7290
9_District1	696	10000	13895602817475.5400	12178217016824.2000	14.1021
9_District2	69	10000	115550871073.1199	89342594098.5600	29.3345
9_District3	637	10000	1136006493400.5298	911115793606.4100	24.6829
9_District4	3600	7739	44898435112269.3300	37164202673658.1000	20.8109
9_District5	1828	10000	30917243479466.1800	25382936280280.4000	21.8032
C101_100t_20w	587	10000	40319250945.5401	-12304901708.1200	427.6682
C101_25t_5w	28	10000	117647561.5199	9512566.6400	1136.7593
C101_50t_10w	102	10000	225963440.8799	-1079154704.1200	120.9389
C102_100t_20w	946	10000	47274199403.2202	-5155408825.8200	1016.9825
C102_25t_5w	31	10000	81101946.7199	-37546115.7600	316.0062
C102_50t_10w	194	10000	1332392101.5199	-1102530180.6200	220.8485
C103_100t_20w	2103	10000	17849419161.1001	-7222438475.8600	347.1384
C103_25t_5w	143	10000	3004716.4199	-43052838.4600	106.9791
C103_50t_10w	518	10000	1367455147.4199	-1110321792.2400	223.1584
C104_100t_20w	3079	10000	10067659651.9603	-8681519205.2400	215.9665
C104_25t_5w	211	10000	72591417.5599	-33540706.8600	316.4278
C104_50t_10w	1017	10000	705156216.5798	-1086946666.3200	164.8749
C105_100t_20w	975	10000	-4085416946.8404	-12158993843.3000	66.4000
C105_25t_5w	33	10000	154693904.5799	1002190.7200	15335.5754
C105_50t_10w	211	10000	-424647555.8200	-1102529614.6600	61.4842
C106_100t_20w	1356	10000	3185665888.6002	-5349954009.7600	159.5456
C106_25t_5w	41	10000	113642696.9800	9512566.6400	1094.6586
C106_50t_10w	177	10000	-319459112.7600	-1086946589.7000	70.6094
C107_100t_20w	1560	10000	3453162979.8198	-6979258964.6200	149.4775
C107_25t_5w	41	10000	149187096.0601	-2001420.3200	7554.0612
C107_50t_10w	355	10000	-420751477.5000	-1125905161.9000	62.6299
C108_100t_20w	1962	10000	18092599436.6800	-6492899831.1000	378.6520
C108_25t_5w	54	10000	113142104.8999	-8009072.9800	1512.6741
C108_50t_10w	425	10000	-467502135.0000	-1125905177.3200	58.4776
C109_100t_20w	2410	10000	10019023325.5600	-6614488230.8200	251.4708
C109_25t_5w	67	10000	43054615.9599	-26031566.9600	265.3938
C109_50t_10w	491	10000	-471398129.1001	-1125905761.1800	58.1316
C201_100t_20w	3600	6740	-12596717125.0994	-12596720162.2300	0.0000
C201_25t_5w	191	10000	-42552163.2400	-45555916.9200	6.5935
C201_50t_10w	999	10000	-1110321196.6001	-1125905241.3800	1.3841
C202_100t_20w	3601	3755	24488233471.3203	-6055175633.1600	504.4182
C202_25t_5w	808	10000	-42552087.8800	-45555950.8800	6.5937
C202_50t_10w	2322	10000	101296080.1398	-1125905361.9600	108.9968
C203_100t_20w	3600	3059	17119879669.9002	-6930623788.8800	347.0178
C203_25t_5w	3600	5635	-41550729.3000	-45555956.2400	8.7918
C203_50t_10w	3601	5719	693469148.1199	-1125905456.0600	161.5921
C204_100t_20w	3602	1746	9897433848.9803	-8657198950.2400	214.3260
C204_25t_5w	3600	4339	-6506840.2200	-45555977.0000	85.7168
C204_50t_10w	3603	3880	93504303.7599	-1125905585.1400	108.3048
C205_100t_20w	3600	4116	9702889586.1598	-12596718687.8400	177.0271
C205_25t_5w	534	10000	-42552150.2200	-45555926.8000	6.5936
C205_50t_10w	2524	10000	-1125904742.5000	-1125905403.9800	0.0000
C206_100t_20w	3600	2927	24536869302.6398	-7733117210.9600	417.2959
C206_25t_5w	713	10000	-41050258.7400	-45555926.8000	9.8904
C206_50t_10w	3434	10000	-498668976.0001	-1125905541.4200	55.7095
C207_100t_20w	3601	1992	9702889889.1399	-7222438653.2400	234.3436
C207_25t_5w	604	10000	-41050268.2199	-45555933.4200	9.8904
C207_50t_10w	3600	3363	-506460328.6400	-1125905455.9000	55.0175
C208_100t_20w	3600	2458	24536869467.2000	-8657201177.3800	383.4272
C208_25t_5w	1142	10000	-42552098.4800	-45555933.1600	6.5937

Table E.9 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
C208_50t_10w	3600	6254	-502564809.1601	-1125905465.6600	55.3634
hh_00_P0	763	10000	51273269138.6235	6663651008.2900	669.4470
ll1_00_P0	171	10000	3750725747.0396	1338732826.5000	180.1698
ll1_01_P0	169	10000	3812991346.5598	1338732826.5000	184.8209
ll1_02_P0	164	10000	2256800886.2898	1338732826.5000	68.5773
ll1_03_P0	161	10000	467118457.0199	1338732826.5000	186.5938 *
ll1_04_P0	169	10000	3781852284.7701	1338732826.5000	182.4949
ll1_05_P0	175	10000	3735116268.2199	1307660206.1200	185.6335
ll1_06_P0	165	10000	7003190961.7999	1214281948.4400	476.7351
ll1_07_P0	173	10000	2132283567.1999	1338732826.5000	59.2762
ll2_00_P0	174	10000	-141721162.5901	85352916.2700	160.2259 *
ll3_00_P0	169	10000	-179672936.1900	-198615845.6200	9.5374
R101_100t_20w	68	10000	21056689632.6198	6319981502.8600	233.1764
R101_25t_5w	6	10000	60631864.9800	47893886.5800	26.5962
R101_50t_10w	17	10000	1308150125.0399	709051142.3992	84.4930
R102_100t_20w	114	10000	19278773781.0998	14101742003.1800	36.7120
R102_25t_5w	9	10000	38715862.5599	34655177.0000	11.7173
R102_50t_10w	24	10000	1161405820.8398	641955625.9388	80.9168
R103_100t_20w	176	10000	13572150034.9598	12469734030.7400	8.8407
R103_25t_5w	14	10000	34265898.6199	30260803.3400	13.2352
R103_50t_10w	38	10000	896486387.7799	570531480.7600	57.1318
R104_100t_20w	227	10000	12739934250.7797	14023383931.9600	10.0742 *
R104_25t_5w	12	10000	25922083.5799	29982663.2800	15.6645 *
R104_50t_10w	66	10000	628969683.6999	293491458.8800	114.3059
R105_100t_20w	110	10000	18503299993.5198	8424839701.0800	119.6279
R105_25t_5w	10	10000	51843140.0799	35155829.6400	47.4666
R105_50t_10w	19	10000	1166167252.9999	388290958.0600	200.3333
R106_100t_20w	145	10000	15976930101.7599	8443754034.7200	89.2159
R106_25t_5w	9	10000	38715863.2599	30483330.4600	27.0066
R106_50t_10w	24	10000	1024184419.4799	303880567.4400	237.0351
R107_100t_20w	184	10000	15206860146.6798	11634816307.3400	30.7013
R107_25t_5w	14	10000	38437663.0399	22028382.6800	74.4915
R107_50t_10w	37	10000	695199501.3999	372274723.4800	86.7436
R108_100t_20w	244	10000	11067396400.7598	12369760088.0800	11.7675 *
R108_25t_5w	12	10000	30093946.9200	26033406.5200	15.5974
R108_50t_10w	61	10000	566635760.1798	293491505.9600	93.0671
R109_100t_20w	187	10000	15204157945.8798	6636116061.1200	129.1122
R109_25t_5w	9	10000	43054585.6799	26255954.0600	63.9802
R109_50t_10w	31	10000	967910656.2599	631133985.4200	53.3605
R110_100t_20w	175	10000	17662978241.2999	7622345982.8200	131.7262
R110_25t_5w	7	10000	43332715.7599	34488424.6600	25.6442
R110_50t_10w	30	10000	905143876.6799	636761297.5800	42.1480
R111_100t_20w	174	10000	15236582303.7799	8473476077.2600	79.8150
R111_25t_5w	12	10000	34432692.7800	26144634.7600	31.7007
R111_50t_10w	30	10000	697796635.3999	496942686.1600	40.4179
R112_100t_20w	211	10000	15239284136.0399	10851236130.0600	40.4382
R112_25t_5w	9	10000	38270855.9599	26311467.9000	45.4531
R112_50t_10w	28	10000	763593813.2199	427249792.1800	78.7230
R201_100t_20w	1951	10000	-575519978.3602	-1383419122.5000	58.3987
R201_25t_5w	98	10000	-5004470.0601	-5171519.4600	3.2301
R201_50t_10w	617	10000	-124231547.3600	-125097475.6400	0.6922
R202_100t_20w	2470	10000	3545030507.4597	-778169509.0800	555.5601
R202_25t_5w	123	10000	-610076.2000	-5171550.1800	88.2032
R202_50t_10w	509	10000	9960054.3198	-125097689.9000	107.9618
R203_100t_20w	2851	10000	1088912826.4598	-891653653.5400	222.1228
R203_25t_5w	315	10000	-554665.9200	-5171614.5200	89.2748
R203_50t_10w	1015	10000	9959621.2399	-125097805.5400	107.9614
R204_100t_20w	3600	8056	259398359.3397	-1007840112.0800	125.7380
R204_25t_5w	169	10000	-721628.9600	-5060450.1000	85.7398
R204_50t_10w	2026	10000	-121201309.0201	-125098025.1800	3.1149
R205_100t_20w	3455	10000	1072700433.2998	-910567724.8000	217.8056
R205_25t_5w	170	10000	-721573.5400	-5171759.2000	86.0478
R205_50t_10w	1120	10000	-122932436.1001	-125097876.3800	1.7309
R206_100t_20w	3600	9881	2720920179.3398	-964607487.1800	382.0753
R206_25t_5w	221	10000	-721607.2800	-5171813.9000	86.0473
R206_50t_10w	844	10000	-55837292.5400	-125097837.4400	55.3651
R207_100t_20w	3600	9582	286418650.8398	-1018647805.1000	128.1175
R207_25t_5w	313	10000	-4837432.8600	-5171794.9600	6.4651
R207_50t_10w	1308	10000	-58001404.1200	-125097967.7600	53.6352
R208_100t_20w	3601	7620	-564711200.2401	-1042965700.1000	45.8552
R208_25t_5w	260	10000	-443556.9401	-5060560.5800	91.2350
R208_50t_10w	1937	10000	-56702907.8401	-125098091.9200	54.6732
R209_100t_20w	3600	9560	1902214701.5199	-1013243421.1000	287.7352
R209_25t_5w	184	10000	-109586.0200	-5060483.0600	97.8344
R209_50t_10w	1477	10000	-119469995.2000	-125097999.8800	4.4988
R210_100t_20w	3424	10000	2720921136.8598	-978117773.2200	378.1792
R210_25t_5w	217	10000	-721602.0800	-5171642.3200	86.0469
R210_50t_10w	1046	10000	-58867197.9600	-125097934.6800	52.9431
R211_100t_20w	3600	9228	5206761206.6198	-942991911.9800	652.1533
R211_25t_5w	156	10000	-721316.9600	-5060502.8400	85.7461
R211_50t_10w	1292	10000	-56270287.9200	-122500836.0400	54.0653
RC101_100t_20w	77	10000	23466874373.1198	15814809991.8800	48.3854
RC101_25t_5w	10	10000	39438826.3999	27034405.3800	45.8838
RC101_50t_10w	22	10000	980031178.0799	190034189.6200	415.7130
RC102_100t_20w	98	10000	19351728427.7198	15822916717.4800	22.3019
RC102_25t_5w	12	10000	39550117.2999	26645148.4800	48.4327
RC102_50t_10w	19	10000	982195906.6799	578756222.9600	69.7080
RC103_100t_20w	136	10000	16914524738.3797	17527878844.9600	3.6261 *
RC103_25t_5w	14	10000	35044648.4199	21805915.8800	60.7116
RC103_50t_10w	28	10000	771385709.1599	767922882.8400	0.4509
RC104_100t_20w	182	10000	13634296649.3597	18130424572.1200	32.9766 *
RC104_25t_5w	14	10000	29982778.9999	25810879.2400	16.1633
RC104_50t_10w	35	10000	573994861.8199	498674346.2200	15.1041
RC105_100t_20w	91	10000	20219070482.6198	10907978098.4200	85.3603
RC105_25t_5w	12	10000	39272067.1599	30761388.4200	27.6667
RC105_50t_10w	23	10000	1234994796.6599	448893536.2400	175.1197
RC106_100t_20w	123	10000	13658614340.6998	11529438265.0600	18.4673

Table E.9 – continued from previous page

Instance	Time	Iterations	Objective Value	Best(SolGH)	Gap
RC106_25t_5w	13	10000	35433819.8199	22417682.5200	58.0619
RC106_50t_10w	20	10000	777445722.5799	515989390.5800	50.6708
RC107_100t_20w	139	10000	18592466464.3797	16638920438.6600	11.7408
RC107_25t_5w	11	10000	39105229.9399	30761589.8800	27.1235
RC107_50t_10w	24	10000	720306398.2399	575726019.2200	25.1127
RC108_100t_20w	156	10000	16852378461.7198	14823176358.6800	13.6893
RC108_25t_5w	12	10000	35322802.2799	30372205.5600	16.2997
RC108_50t_10w	27	10000	772251501.7799	630268451.2000	22.5273
RC201_100t_20w	1822	10000	-1399627245.4002	-1378013035.7800	1.5442 *
RC201_25t_5w	88	10000	-5003835.8200	-5171005.4000	3.2328
RC201_50t_10w	403	10000	-123363742.0002	-125096456.0400	1.3851
RC202_100t_20w	2096	10000	1902215724.7197	-848419159.5200	324.2070
RC202_25t_5w	100	10000	-721207.8400	-5171480.4200	86.0541
RC202_50t_10w	404	10000	15588209.9998	-125096955.0600	112.4609
RC203_100t_20w	2603	10000	-559306081.1203	-915968950.1800	38.9383
RC203_25t_5w	206	10000	-721475.4200	-5171505.3800	86.0490
RC203_50t_10w	700	10000	77055884.3598	-125096765.1800	161.5970
RC204_100t_20w	3600	9231	-559305976.1602	-1032155821.9000	45.8118
RC204_25t_5w	161	10000	-554372.4600	-5060413.3600	89.0449
RC204_50t_10w	1291	10000	9094797.9999	-125097014.9000	107.2701
RC205_100t_20w	2182	10000	1099722278.2797	-856525543.8000	228.3934
RC205_25t_5w	90	10000	-5059843.1600	-5171462.0600	2.1583
RC205_50t_10w	524	10000	11260334.2598	-125096766.4200	109.0012
RC206_100t_20w	2783	10000	-556603315.8802	-905161565.1000	38.5078
RC206_25t_5w	123	10000	-5004067.1400	-5171397.9400	3.2356
RC206_50t_10w	716	10000	-120766725.9201	-125097137.7600	3.4616
RC207_100t_20w	2577	10000	1102424195.3998	-907864881.6600	221.4304
RC207_25t_5w	185	10000	-4447914.4400	-5059890.2000	12.0946
RC207_50t_10w	719	10000	-119468826.4401	-122499526.2000	2.4740
RC208_100t_20w	3200	10000	2734432112.6797	-978116228.2000	379.5610
RC208_25t_5w	182	10000	-4726710.9000	-5060744.0000	6.6004
RC208_50t_10w	909	10000	-56268580.4801	-125096819.1800	55.0199
test150-0-0-0-0_d0_tw0	3600	4056	6897664625.3999	-28349336446.9000	124.3309
test150-0-0-0-0_d0_tw1	3600	7802	-55870971772.2999	-30832491493.7000	44.8148 *
test150-0-0-0-0_d0_tw2	3600	8672	-55870971432.3001	-28832172176.8000	48.3950 *
test150-0-0-0-0_d0_tw3	3600	9460	-55870971746.7008	-27383664735.0000	50.9876 *
test150-0-0-0-0_d0_tw4	3600	9108	101050622709.4000	-26142085667.0000	486.5438
test250-0-0-0-0_d0_tw0	-	-	-	23650649218.5000	-
test250-0-0-0-0_d0_tw1	-	-	-	-292254131472.0000	-
test250-0-0-0-0_d0_tw2	3600	2546	-214544937458.1017	-275698693917.1000	22.1813
test250-0-0-0-0_d0_tw3	3600	2924	-214544933324.4147	-266914175565.0000	19.6202
test250-0-0-0-0_d0_tw4	-	-	-	-231776103927.7000	-
test50-0-0-0-0_d0_tw0	2621	10000	-842598905.0999	-842599324.2000	0.0000
test50-0-0-0-0_d0_tw1	1091	10000	-842598625.1999	-842599506.6000	0.0001
test50-0-0-0-0_d0_tw2	911	10000	-842598189.0999	-842599560.3000	0.0001
test50-0-0-0-0_d0_tw3	735	10000	-842597667.5999	-842598590.0000	0.0001
test50-0-0-0-0_d0_tw4	815	10000	-842598471.6999	-842599753.8000	0.0001